STATISTICAL AND GEOMETRIC METHODS FOR VISUAL TRACKING WITH OCCLUSION HANDLING AND TARGET REACQUISITION

A Thesis
Presented to
The Academic Faculty

by

Jehoon Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2012

STATISTICAL AND GEOMETRIC METHODS FOR VISUAL TRACKING WITH OCCLUSION HANDLING AND TARGET REACQUISITION

Approved by:

Professor Allen Tannenbaum, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Anthony Yezzi, Co-Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Jeff Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Patricio Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yorai Wardi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Sung Ha Kang
School of Mathematics
Georgia Institute of Technology

Date Approved: 19 December 2011

To my family, this work is yours.

It couldn't be done without your love and trust.

ACKNOWLEDGEMENTS

Throughout my Ph.D. time at Georgia Institute of Technology, I have met and worked with a number of great people who need to be thanked for their help to allow me to complete my Ph.D. degree. It is a pleasure to convey my gratitude to them all in my humble acknowledgment. In particular, I would like to express my sincere and deepest appreciation to:

- Professor Allen Tannenbaum for your guidance, support, and encouragement. My research is absolutely inspired by your intuition and knowledge. Not only have I learned a great amount about computer vision from you, but you also have taught me a lot about life, humor, and friendship. It has been truly honor to be your student. I am forever indebted to you.

- Professor Anthony Yezzi for your constant help, kindness, and sincerity since I have met you. Your research has inspired and had valuable influence on my studies. With sincere and deep feeling, I truly respect your excellence in academics and your generous spirit in all aspects of life.

- Professor Jeff Shamma, Professor Patricio Vela, Professor Yorai Wardi, and Professor Sung Ha Kang for accepting to be a member of my Ph.D. thesis committee, and your valuable remarks that helped improve my thesis.

- Research Collaborators: Shawn Lankton, Romeil Sandhu, Liangjia Zhu, Ivan Kolesov, and Peter Karasev for being an inspiring research partner and friend, and your cheerful encouragement and friendship that made wonderful research possible.

- Members of the MINERVA Research Group: Yi Gao, Vandana Mohan, Eli Hershkovits, John Melonakos, Tauseef Rehman, Gallagher Pryor, Jimi Malcolm, Behnood Gholami, Xavier LeFaucheur, Jacob Huang, Martin Mueller, and Yifei Lou for making the lab a joyful and studious atmosphere that helped me tremendously enjoy my research.

Last but not least,

- All my friends in Atlanta and elsewhere, particularly, my dear friends in Korea for the happy and beautiful memories we have shared over the years.

- My father Wonsoo Lee, my mother Soonhyung Lee, my brother Donggun Lee, and all my family for your unconditional love, continuous support, and selfless sacrifices. You always gave me the strength to dream. You kept me always on the right direction. You always stood by me in good or bad times. I owe you everything precious in my life.

- My wife Hyorim Lee, and my baby Julian Huije Lee. There are no words that express my thankfulness to you. You are my everything. You are the reason I am. From the bottom of my heart, I love you so much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Computer vision is the science that studies how machines understand scenes and automatically make decisions based on meaningful information extracted from an image or multi-dimensional data of the scene, like human vision. One common and well-studied field of computer vision is visual tracking. It is challenging and active research area in the computer vision community. Visual tracking is the task of continuously estimating the pose of an object of interest from the background in consecutive frames of an image sequence. It is a ubiquitous task and a fundamental technology of computer vision that provides low-level information used for high-level applications such as visual navigation, human-computer interaction, and surveillance system.

The focus of the research in this thesis is visual tracking and its applications. More specifically, the object of this research is to design a reliable tracking algorithm for a deformable object that is robust to clutter and capable of occlusion handling and target reacquisition in realistic tracking scenarios by using statistical and geometric methods. To this end, the approaches developed in this thesis make extensive use of region-based active contours and particle filters in a variational framework. In addition, to deal with occlusions and target reacquisition problems, we exploit the benefits of coupling 2D and 3D information of an image and an object.

In this thesis, first, we present an approach for tracking a moving object based on 3D range information in stereoscopic temporal imagery by combining particle filtering and geometric active contours. Range information is weighted by the proposed Gaussian weighting scheme to improve segmentation achieved by active contours. In addition, this work present an on-line shape learning method based on principal component analysis to reacquire track of an object in the event that it disappears from the field of view and reappears later. Second, we propose an approach to jointly track a rigid object in a 2D image sequence and to estimate its pose in 3D space. In this work, we take advantage of knowledge of a 3D model of an object and we employ particle filtering to generate

and propagate the translation and rotation parameters in a decoupled manner. Moreover, to continuously track the object in the presence of occlusions, we propose an occlusion detection and handling scheme based on the control of the degree of dependence between predictions and measurements of the system. Third, we introduce the fast level-set based algorithm applicable to real-time applications. In this algorithm, a contour-based tracker is improved in terms of computational complexity and the tracker performs real-time curve evolution for detecting multiple windows. Lastly, we deal with rapid human motion in context of object segmentation and visual tracking. Specifically, we introduce a model-free and marker-less approach for human body tracking based on a dynamic color model and geometric information of a human body from a monocular video sequence. The contributions of this thesis are summarized as follows:

- Reliable algorithm to track deformable objects in a sequence consisting of 3D range data by combining particle filtering and statistics-based active contour models.
- Effective handling scheme based on object's 2D shape information for the challenging situations in which the tracked object is completely gone from the image domain during tracking.
- Robust 2D-3D pose tracking algorithm using a 3D shape prior and particle filters on SE(3).
- Occlusion handling scheme based on the degree of trust between predictions and measurements of the tracking system, which is controlled in an online fashion.
- Fast level set based active contour models applicable to real-time object detection.
- Model-free and marker-less approach for tracking of rapid human motion based on a dynamic color model and geometric information of a human body.

CHAPTER I

INTRODUCTION

In this chapter, we introduce the definitions of computer vision and visual tracking with their various applications. We describe several difficult problems in visual tracking, and propose methods to solve them. Contributions of this thesis and a brief summary of the contents of the remaining chapters are included at the end of this chapter.

1.1 Computer Vision and Visual Tracking

Computer vision is the science that studies how machines see (or understand) scenes and automatically make decisions based on meaningful information extracted from an image or multi-dimensional data of the scene, like human vision [41]. The design of a computer vision system strongly depends on its application and intended functionality, but it is generally composed of three processes: image acquisition to obtain data, image processing to extract useful information, and image understanding and interpretation to use this information for making higher level decisions.

One common and well-studied field of computer vision is visual tracking, which is challenging and active research area in the computer vision community. Visual tracking is the task of continuously estimating the pose (position and orientation) of an object of interest from the background in consecutive frames of an image sequence. A large number of tracking algorithms have inspired significant interest in recent years (see [9, 8, 42, 123], references therein, and literature reviewed in each chapter of this thesis). The motion of an object is observed using an image input device (e.g., a digital camera or a 3D scanner) that generates a sequence of images at discrete time steps. And the task of tracking refers to state estimation of the relative camera-target pose from an image sequence. It is a ubiquitous task and a fundamental technology of computer vision that provides low-level information used for high-level applications that include:

- Vehicle guidance and control: vision-based path planning and obstacle avoidance for an Unmanned Ground Vehicle (UGV). The automated landing system for an Unmanned Aerial

Vehicle (UAV).

- Visual navigation: visual SLAM (Simultaneous Localization and Mapping) for autonomous robots.
- Motion capture: tracking and recording human movements later to be used for rendering process of an animated character or medical analysis of a subject's movement.
- Medical instrument: vision guided or tele-operative surgery for treatment.
- Human-computer interaction: systems recognizing people's actions (or gestures) to drive computer devices, and to provide the user with feedback from the computer.
- Surveillance system: automated surveillance system monitoring a scene to detect suspicious activities or abnormal motion pattern.
- Traffic control: traffic monitoring system gathering traffic statistics to control traffic flow.
- Manufacturing: vision control system to allow a robot arm to pick up and deliver a designated object by measuring its position and orientation.

The overall motion of an object can be described by the combination of global motion of the object and its local deformations [121]. Therefore, the objective of object tracking is to estimate a temporal function describing the group displacement and shape deformation of an object. Tracking a deformable object can be accomplished with a contour-based tracker [77, 88] in which object segmentation is carried out by active contours [52]. Image (or object) segmentation is the task of clustering pixels into salient image regions corresponding to meaningful objects or regions in static imagery. This task usually leads to the separation an object of interest from the background in an image. Active contours provide a robust framework to address the problem of image segmentation; they are based on variational methods in which the closed curve evolves or deforms so as to minimize a defined energy functional until they delineate the borders of an object of interest. In the contour-based tracker, active contours begin their evolution at the initial position predicted by the motion model and evolve to segment the object of interest. Upon convergence, the object is

represented by the closed contours. Thus, the contour-based tracker is suitable for tracking the object with complex shape. The extracted shape information of the tracked object is further used for advanced applications, such as shape learning and classification.

To accurately estimate the position of an object of interest, reliable measurement is necessary. In the contour-based tracker, segmentation results commonly obtained by active contours usually serve as a measurement model. However, active contours often lead to poor segmentation results because of noise, various photometric artifacts, clutter, etc; thus, it is not guaranteed to converge to the exact boundary of an object. To remedy this problem, 3D range data (or depth maps), computed by determining correspondences from stereo sequences, can be used to improve the quality of the segmentation task via active contours. Such a sensor-fusion approach provides more robust state estimate in a sequence of images as well as leads to robust segmentation in a single image. In addition, a 3D shape prior can be used to improve tracking performance. In particular, the knowledge of a 3D model allows the tracker to capture various aspects of the tracked object with respect to its dynamic motion than using a collection of 2D shape priors and to be extended to jointly estimate object's 2D and 3D positions in a single framework, which is also known as the 2D-3D pose tracking problem.

In a real-world sequence, several challenges arise, such as scene illumination changes, changes in the objects size or appearance, and rapid or complex object motion. These problems of tracking can be simplified by imposing constraints on the motion of the object of interest and defining its appearance model. For example, tracking algorithms usually assume that the motion of the object is smooth with no abrupt or erratic changes and follows simple dynamics, such as constant velocity or acceleration. Further assumptions, such as the number and the size of the object, can also be adopted to simplify the problem. However, object tracking becomes more challenging when the object of interest is in the presence of noise and clutter because it may not appear as the assumed model. Moreover, it is particularly difficult to effectively handle the challenging situations in which the tracked object is partially occluded or completely gone from the image domain during tracking. Such a complex situation is challenging (but common) in realistic tracking scenarios. In general, to solve these problems, a filtering scheme (e.g., Kalman filters or particle filters) is incorporated to effectively treat the dynamic nature of the observed 2D scene and to increase the robustness of tracking. This scheme makes use of information about the dynamics of the object learned on-line or

off-line. In addition, one can take advantage of the object's shape information. Shape information provides necessary constraints for maintaining track when the object being tracked is occluded in a cluttered environment or disappears completely. The additional information is used to make up for poorly distinguishable statistics between the object and background or missing parts of the object.

1.2 Contributions and Organization of this Thesis

The focus of the research in this thesis is visual tracking and its applications. More specifically, the object of this research is to design a reliable tracking algorithm for a deformable object that is robust to clutter and capable of occlusion handling and target reacquisition in realistic tracking scenarios by using statistical and geometric methods. To this end, the approaches developed in this thesis make extensive use of region-based active contours and particle filters in a variational framework. In addition, to deal with occlusions and target reacquisition problems, we exploit the benefits of coupling 2D and 3D information (3D range data or a 3D shape prior) and supplementary information such as object's shape in 2D. The contributions of the present work and the problems considered here are as follows:

- Reliable algorithm to track deformable objects in a sequence consisting of 3D range data by combining particle filtering and statistics-based active contour models.
- Effective handling scheme based on object's 2D shape information for the challenging situations in which the tracked object is completely gone from the image domain during tracking.
- Robust 2D-3D pose tracking algorithm using a 3D shape prior and particle filters on SE(3).
- Occlusion handling scheme based on the degree of trust between predictions and measurements of the tracking system, which is controlled in an online fashion.
- Fast level set based active contour models applicable to real-time object detection.
- Model-free and marker-less approach for tracking of rapid human motion based on a dynamic color model and geometric information of a human body.

This thesis is divided into five main chapters that are meant to be relatively self-contained. The followings are brief summaries and contributions of the remaining chapters of this thesis:

- Chapter 2: we briefly introduce some of the basic ideas for geometric active contours with level set methods and Bayesian state estimation with particle filtering. These ideas are extensively used to design tracking frameworks proposed in this thesis.

- Chapter 3: we propose an approach for tracking an object of interest based on 3D range data. We employ particle filtering and active contours to simultaneously estimate the global motion of the object and its local deformations. The proposed algorithm takes advantage of range information to simplify and improve the segmentation process via active contours. The tracking framework proposed in this chapter deals with the challenging (but common) situation in which the tracked object disappears from the image domain entirely and reappears later. To cope with this problem, a method based on principle component analysis (PCA) of shape information is proposed. In the proposed method, if the target disappears out of frame, shape similarity energy is used to detect target candidates that match a template shape learned online from previously observed frames. Thus, we require no priori knowledge of the target's shape. Moreover, to simplify the tracking framework and to improve the computational efficiency, an alternative tracking framework using dynamic weighted depth maps is proposed; the weighted depth maps are dynamically generated based on the previous tracking results. In this approach, the global motion of the tracked object can be estimated by active contour segmentation using the proposed dynamic weighting scheme. Experimental results of both frameworks show the practical applicability and robustness of the proposed algorithms in realistic tracking scenarios.

- Chapter 4: we present a visual pose tracking algorithm based on Monte Carlo sampling of special Euclidean group SE(3) and knowledge of a 3D model. In general, the relative pose of an object in 3D space can be described by sequential application of transformation matrices at each time step. Thus, the objective of this work is to find a transformation matrix in SE(3) so that the projection of an object transformed by this matrix coincides with an object of interest in the 2D image plane. To do this, first, the set of these transformation matrices is randomly generated via an autoregressive model. Next, 3D transformation is performed on a 3D model by these matrices. Finally, a region-based energy model is designed to evaluate the optimality

of a transformed model's projection. In order to further improve the performance of 2D-3D pose tracking and occlusion handling, we propose a new filtering based tracking framework. In particular, we revisit a joint 2D segmentation/3D pose estimation technique of [25], and extend it by incorporating particle filters to track the object and its corresponding pose. Moreover, to allow the algorithm to continuously track the object in the presence of occlusions, the following scheme is developed. First, a histogram-based appearance model is created and updated to detect occlusions. Second, a dynamical choice of how to invoke the objective functional is performed online to handle the occlusion. The decision is based on the degree of occlusion and the variation of the object's pose to control the degree of dependence between predictions and measurements of the system. Experimental results demonstrate the practical applicability and robustness of the proposed methods in several challenging scenarios.

- Chapter 5: we present an algorithm for the detection of windows using active contours. The proposed algorithm provides a robust and fast segmentation approach for real-time applications. Active contours implemented in the level set framework have naturally high computational complexity since the curve is implicitly represented by a higher dimensional function. The tracker performs real-time curve evolution for detecting windows by combining two fast algorithms for level set methods proposed by Song and Chan [107] and Shi and Karl [105] in the framework of the Chan-Vese active contour model [13]. The proposed method is designed to detect multiple windows fast enough for an unmanned aerial vehicle (UAV) to carry out missions in the 2008 International Aerial Robotics Competition (IARC).

- Chapter 6: we deal with rapid human motion in context of object segmentation and visual tracking. Specifically, we introduce a model-free and marker-less approach for human body tracking based on a dynamic color model and geometric information of a human body from a monocular video sequence. A multivariate Gaussian distribution is learned online from sequential frames to represent the non-stationary color distribution. Images are first filtered according to the current color model allowing a human body to be segmented from a background. Next, the segmented image is partitioned based on geometric prior knowledge of the human skeleton and each body part of interest is separately tracked. Finally, eigen-axis

based joint angle estimation is carried out to evaluate jumping and landing posture. The resulting data facilitates motion analysis of a specific set of clinically interesting quantities; then, post-operative evaluation and correlation of injury statistics with a subject's mechanics can be carried out. The proposed approach is tested on different views of human jumping and landing sequences in a noisy and cluttered environment with video from a mobile-phone camera. Experimental results demonstrate the practical applicability and robustness of the proposed algorithm in tracking human motion captured with a monocular camera system.

CHAPTER II

PRELIMINARIES

The works of this thesis are inspired by contour-based tracking frameworks and non-linear filtering schemes. Thus, in this chapter, we briefly introduce some of the basic ideas for geometric active contours with level set methods and Bayesian state estimation with particle filtering. These preliminaries will be cornerstone of this thesis and they will be extensively used in the works of this thesis.

2.1 Geometric Active Contours and Level Set Methods

Since the introduction of deformable contours or snakes by Kass, Witkin, and Terzopoulos [52], curve evolution has been widely used for segmentation in single images as well as visual tracking in image sequences. The basic idea of active contour models is to evolve or deform the closed curve so as to minimize an energy functional by gradient descent until it delineates the borders of an object of interest in an image (see [52, 111] for details). Thus, curve evolution is driven by a gradient flow of an energy functional, which depends on either local or global properties of the image and external constraints, such as smoothness or shape. Active contours are generally categorized into two types; edge-based and region-based. In the former, the contour evolves in response to local image features such as image gradient while in the latter, the contour evolves in response to regional image statistics. Since only information located close to the evolving contour is examined in edge-based models, edge-based models are susceptible to noise, weak edges, and missing information. On the other hand, compared to edge-based models, region-based approaches are robust to noise because it utilizes more global information of an image.

In region-based models, the evolution of the curve depends on region statistics. Many of the region-based models have been inspired by the region competition work of Zhu and Yuille [126]. In this model, an energy $E(C)$ is defined as a particular function f over a region R inside the curve C:

$$E(C) = \int_R f(\mathrm{x}) d\mathrm{x}. \tag{1}$$

Using the divergence theorem and a fixed parameterization $p \in [0,\ 1]$ of the curve C, which is independent of time t as the curve evolves, we rewrite (1) as

$$E(C) = \oint_C < \mathbf{F},\ \mathbf{N} > ds = \int_0^1 < \mathbf{F},\ JC_p > dp \tag{2}$$

where $\mathbf{F}$ is a vector field chosen so that $\nabla \cdot \mathbf{F}(\mathbf{x}) = f(\mathbf{x})$. $\mathbf{N}$ denotes the unit normal of C and ds is the Euclidean arc length element. $J = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ is the direct $\frac{\pi}{2}$-rotation matrix. Now differentiating E with respect to t yields

$$\frac{dE}{dt} = \int_0^1 \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_t,\ JC_p \right\rangle + \langle \mathbf{F},\ JC_{pt} \rangle\, dp \tag{3}$$

where $\frac{d\mathbf{F}}{d\mathbf{x}}$ denotes the Jacobian matrix of $\mathbf{F}$ with respect to $\mathbf{x}$. By using the integration by parts and changing dp into ds, (3) becomes:

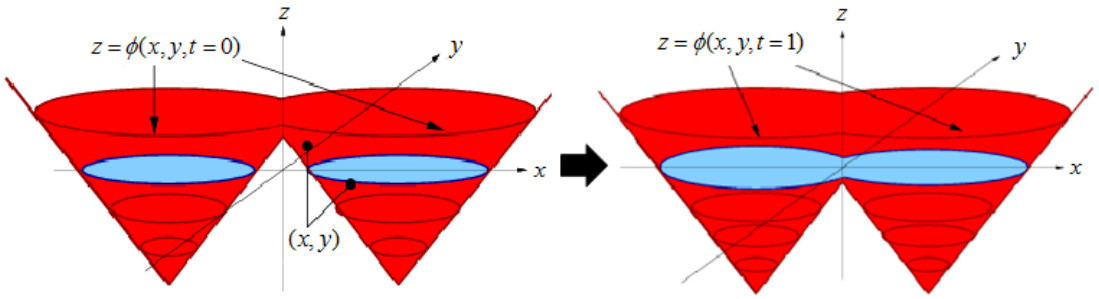
$$\begin{aligned} \frac{dE}{dt} &= \int_0^1 \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_t,\ JC_p \right\rangle - \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_p,\ JC_t \right\rangle dp \\ &= \oint_C \left\langle C_t,\ \left(\left(\frac{d\mathbf{F}}{d\mathbf{x}}\right)^T J - J^T \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) \right) C_s \right\rangle ds \\ &= \oint_C \langle C_t,\ f\mathbf{N} \rangle\, ds. \end{aligned} \tag{4}$$

From (4), one can derive the gradient flow of C that minimizes the energy $E(C)$ in (1) (see Appendix A for details):

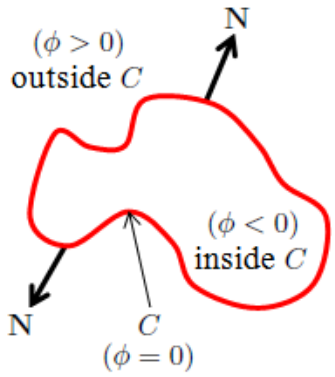
$$\frac{\partial C}{\partial t} = -\nabla_C E = -f\mathbf{N}. \tag{5}$$

For region-based active contours, many separation energies between inside and outside of the curve have been proposed to drive the segmenting curve toward the boundaries of an object. For example, distinct mean intensities [13, 120], Gaussian distributions [81, 23], and intensity histograms [73, 53] are generally used for the objective energy.

To implement active contours, the closed curve can be represented explicitly [52] or implicitly [12]. The explicitly represented curve is evolved by updating the corresponding parameters or control points while the implicitly represented curve is evolved via the level set methods introduced by Osher and Sethian [78, 102]. The level set methods offer a powerful representation tool for the numerical implementation of curve evolution. This method can automatically handle topological changes such as curve splitting and merging, with which parametric representations have difficulty



(a)



(b)

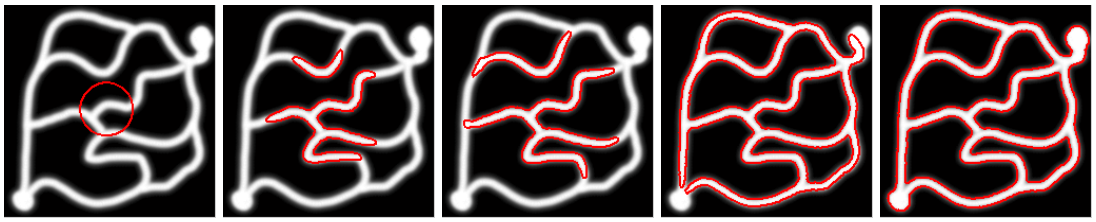
Figure 1: (a) Level sets of an embedding function ϕ. (b) The closed curve C in $\mathbb{R}^2$. The curve C is implicitly represented as a level set of ϕ. Figures are obtained from [102].

in handling. Using level set methods makes a flow of the contours to be *geometric* because their curve evolution depends only on geometric properties of the contour, and is independent of parameterization. In the level set methods, a closed curve C is represented as the zero level set of a higher dimensional function ϕ, which is typically chosen to be a signed distance function such that $\phi < 0$ inside C and $\phi > 0$ outside C. Therefore, the curve can be described by an implicit surface as (see Figure 1.)

$$C = \{\mathrm{x} \mid \phi(\mathrm{x}) \equiv 0,\ \mathrm{x} \in \Omega\} \tag{6}$$

where Ω is the domain of an image $I(\mathrm{x})$: $\mathbb{R}^2 \rightarrow \mathbb{Z}$ which is mapping to the photometric variable $z \in \mathbb{Z}$. By doing so, the curve C propagates implicitly as the level set function ϕ evolves explicitly. The curve C is embedded as the zero level set such that $\phi(C, t) = 0$. Now, differentiating ϕ with

(a)



(b)

Figure 2: Segmentation results of (a) the airplane, and (b) the cavern using the region-based active contours implemented by the level set methods. The region-based energy is based on [13]. Left to right: initial, intermediate and final contours.

respect to t using the chain-rule, we have

$$\begin{aligned} \frac{\partial}{\partial t}\phi(C,t) &= 0 \\ \left\langle \nabla_C \phi, \frac{\partial C}{\partial t} \right\rangle + \frac{\partial \phi}{\partial t} &= 0 \\ \langle \nabla_C \phi, \nu_{\mathbf{N}} \mathbf{N} \rangle + \frac{\partial \phi}{\partial t} &= 0 \\ \nu_{\mathbf{N}} \nabla_C \phi + \frac{\partial \phi}{\partial t} &= 0 \end{aligned} \tag{7}$$

where the unit (outward) normal is defined in terms of ϕ as $\mathbf{N} = \frac{\nabla_C \phi}{\|\nabla_C \phi\|}$, and $\nu_{\mathbf{N}}$ is the normal part of $\frac{\partial C}{\partial t}$. Note that the tangential part of $\frac{\partial C}{\partial t}$ is eliminated and only the normal part remains since $\nabla_C \phi$ is perpendicular to the tangent to $C(t)$ in (7), and the inner product is zero. Finally, the gradient flow of ϕ is

$$\frac{\partial \phi}{\partial t} = -\nu_{\mathbf{N}} \nabla_C \phi. \tag{8}$$

Now, representing the energy $E(C)$ as the level set function ϕ instead of a function of C as $E(\phi)$, we get:

$$\frac{\partial \phi}{\partial t} = \|\nabla_C \phi\| \langle \nabla_C E, \mathbf{N} \rangle . \tag{9}$$

Thus, by substituting (5) into the equation above, i.e., $\nabla_C E = f\mathbf{N}$, the level set flow of the region-based active contours is represented by level set function ϕ as

$$\frac{\partial \phi}{\partial t} = f \|\nabla_C \phi\| . \tag{10}$$

Figure 2 shows the segmentation results using region-based active contours implemented by level set methods.

2.2 Bayesian State Estimation and Particle Filters

The overall objective of state estimation is to estimate a (hidden) variable of interest governing a particular underlying system with respect to what is being observed or measured. With this goal, we let s_t be a (usually hidden or not observable) state vector and z_t be a set of observations. Then, a nonlinear stochastic system can be described by a stochastic discrete-time state space transition (dynamic) equation and the stochastic observation (measurement) equation as

$$\begin{aligned} s_{t+1} &= f_t(s_t, u_t) \\ z_t &= h_t(s_t, v_t) \end{aligned} \tag{11}$$

where t is the time index, and the functions $f_t(\cdot)$ and $h_t(\cdot)$ are a time-varying nonlinear system and a measurement equation, respectively. u_t and v_t are independent and identically distributed (iid) random variables representing noise whose probability density functions (pdfs) are known. Furthermore, we assume that the initial state distribution $p(s_0)$ is known and can be written as

$$\begin{aligned} p(s_0) &= p(s_0|z_0) \\ &= \frac{p(z_0|s_0)}{p(z_0)} p(s_0) \end{aligned} \tag{12}$$

where z_0 is the set of no measurements.

The Bayesian approach is to find estimates of s_t based on all available measurements up to time t, $z_{1:t}$, by approximating the conditional pdf of s_t, $p(s_t|z_t)$. In the Bayesian context, the conditional pdf $p(s_t|z_{1:t})$ is estimated in a recursive manner including the prediction step and the update step. In the prediction step, the *a priori* of the state s_t at time t with knowledge of the measurement z_t up to $t-1$ is obtained by using Chapman-Kolmogorov equation as

$$\begin{aligned} p(s_t|z_{1:t-1}) &= \int p(s_t|s_{t-1}, z_{1:t-1}) p(s_{t-1}|z_{1:t-1}) \, ds_{t-1} \\ &= \int p(s_t|s_{t-1}) p(s_{t-1}|z_{1:t-1}) \, ds_{t-1}. \end{aligned} \tag{13}$$

Note that s_t is entirely determined by s_{t-1} and u_t in (11).

At the time step t, when a new measurement z_t becomes available, we obtain *a posteriori* pdf by Bayes' rule and factorization of joint pdfs as

$$\begin{aligned} p(s_t|z_{1:t}) &= \frac{p(z_{1:t}|s_t)p(s_t)}{p(z_{1:t})} \\ &= \frac{p(z_t, z_{1:t-1}|s_t)p(s_t)}{p(z_t, z_{1:t-1})} \\ &= \frac{p(z_t, z_{1:t-1}|s_t)p(z_{1:t-1}s_t)p(s_t)}{p(z_t, z_{1:t-1})p(z_{1:t-1})} \\ &= \frac{p(z_t, z_{1:t-1}|s_t)p(s_t|z_{1:t-1})p(z_{1:t-1})p(s_t)}{p(z_t, z_{1:t-1})p(z_{1:t-1})p(s_t)} \\ &= \frac{p(z_t|s_t)p(s_t|z_{1:t-1})}{p(z_t|z_{1:t-1})}. \end{aligned} \tag{14}$$

This is an update of the prior pdf $p(s_t|z_{1:t-1})$ using the measurement z_t. Here, the pdf $p(z_t|z_{1:t-1})$ is a normalizing constant as

$$\begin{aligned} p(z_t|z_{1:t-1}) &= \int p(z_t|s_t, z_{t-1})p(s_t|z_{1:t-1})\, ds_t \\ &= \int p(z_t|s_t)p(s_t|z_{1:t-1})\, ds_t \end{aligned} \tag{15}$$

where $p(z_t|s_t, z_{t-1}) = p(z_t|s_t)$ since z_t only depends on s_t and v_t in (11). Now, one can compute an optimal state estimate with some criteria. For example, the minimum mean-square error (MMSE) estimate is the conditional mean of s_t or the maximum a posteriori (MAP) estimate is the maximum of the posterior $p(s_t|z_{1:t})$

The recursive estimation of the posterior density cannot be determined analytically, i.e., the analytic solutions of these equations above are not available except a small restrictive number of cases. For example, if $f_t(\cdot)$ and $h_t(\cdot)$ are linear models with Gaussian noise distributions, Kalman filter [50] produces the optimal state estimate. For nonlinear models, one has to use approximations for the analytic solution of the Bayesian prediction and update equations in (13) and (14), and use nonlinear filters, such as the unscented Kalman filter or the particle filter. Nonlinear filtering is the process of estimating the state of a nonlinear stochastic system from non-Gaussian and noisy observation data. In particular, particle filtering is a popular method to solve nonlinear and/or non-Gaussian Bayesian filtering problem with Monte Carlo Simulation. In the contour-based tracker, the measurement model is highly non-linear due to the effects of the curve evolution and the state density of the system is non-Gaussian. Thus, in the presented works of this thesis, particle filtering [37, 4] is adopted to cope with this non-linearity and non-Gaussian estimation problems.

Sequential Bayesian filtering estimation with Monte Carlo simulation, called *particle filtering*, was first introduced by Gordon, Salmond, and Smith [37]. In recent years, it has proven to be a powerful scheme for non-linear and non-Gaussian estimation problems due to its simplicity and versatility [4].

Drawing $N >> 1$ independent samples $\{s_t^i\}_{i=1,\ldots,N}$ from $p(s_t|z_{1:t})$, one can properly construct an empirical estimate of the true hidden state s_t. However, generating samples from the posterior distribution $p(s_t|z_{1:t})$ is usually not possible (e.g., $f_t(\cdot)$ and $h_t(\cdot)$ are nonlinear). Thus, if one can only generate samples from a similar (or proposal) density $q(s_t|z_{1:t}) \approx p(s_t|z_{1:t})$ instead of the desired distribution, the problem becomes one of "importance sampling." That is, one can form a Monte Carlo estimate of s_t by generating N samples according to $q(s_t|z_{1:t})$ with associated importance weights $\{w_t^i\}_{i=1,\ldots,N}$ at each time t. More importantly, as the algorithm progresses and if N is chosen sufficiently large, the proposal distribution can be shown to "evolve" toward the correct posterior distribution, i.e., $q(s_t|z_{1:t}) = p(s_t|z_{1:t})$.

Thus, the generic algorithm begins by first sampling N times from initial state distribution, $p(s_0)$. Following this, the algorithm can be decomposed in two steps: the prediction step and the update step. Using importance sampling [29], the **prediction step** is the act of drawing N samples from the alternative proposal distribution $q(s_t|z_{1:t})$. As new information arrives online at time t from the observation z_t, one needs to evaluate the "fitness" of the predicted samples or particles. In other words, as z_t becomes available, the measurement or **update step** in particle filtering is incorporated through the importance weights by the following equation

$$w_t^i = w_{t-1}^i \frac{p(z_t \mid s_t^i) p(s_t^i \mid s_{t-1}^i)}{q(s_t^i \mid s_{1:t-1}^i, z_{1:t})} \tag{16}$$

where $p(z_t \mid s_t^i)$ is the likelihood of the arrived observation at time t. Also, w_t^i is the normalized weight of i-th particle or $\sum_{i=1}^N w_t^i = 1$. Furthermore, if $q(s_t^i \mid s_{1:t-1}^i, z_{1:t}) = q(s_t^i \mid s_{t-1}^i, z_t)$, then the important density q depends only on the s_{t-1} and z_t as

$$w_t^i \propto w_{t-1}^i \frac{p(z_t \mid s_t^i) p(s_t^i \mid s_{t-1}^i)}{q(s_t^i \mid s_{t-1}^i, z_t)}. \tag{17}$$

From the above approach, the filtering distribution is represented by a set of samples s_t^i and its

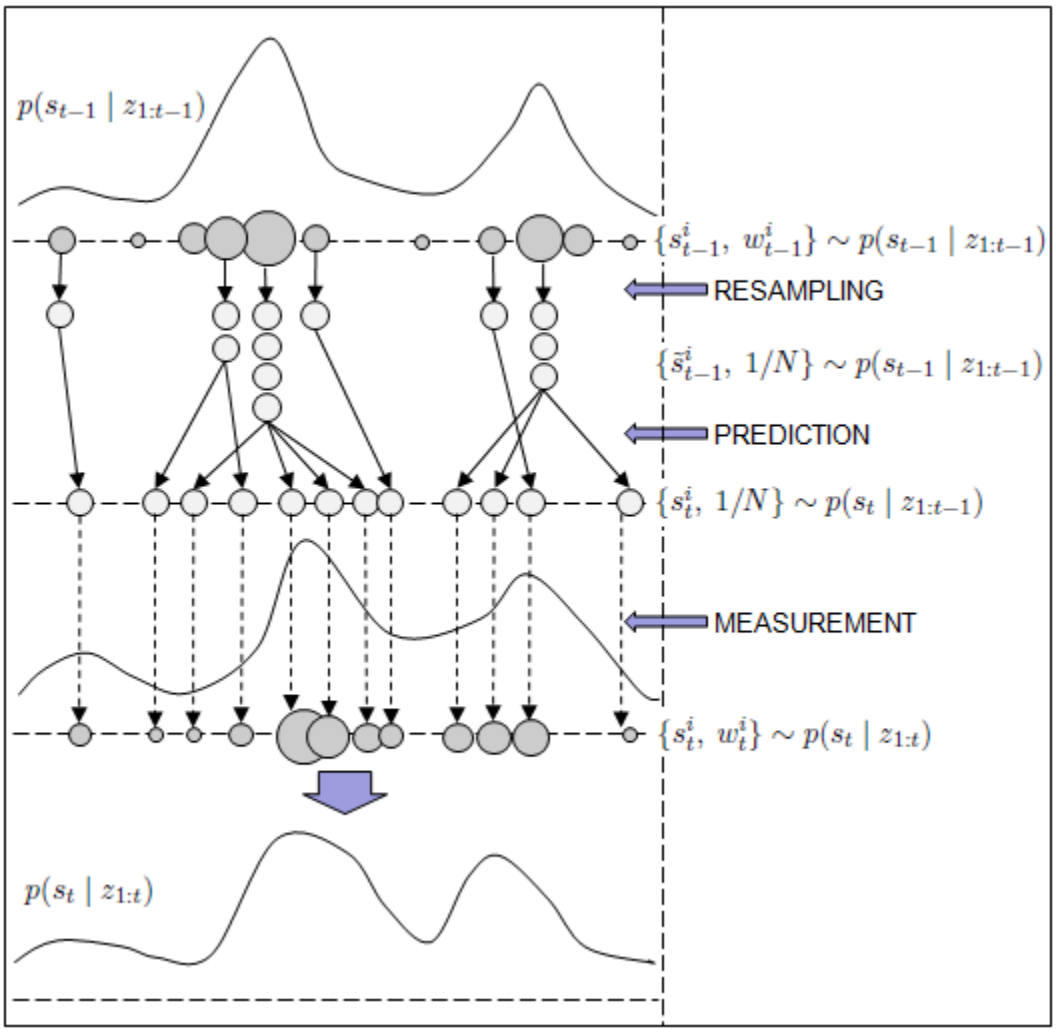


Figure 3: A graphical representation of sequential importance resampling (SIR) particle filters.

associated weights w_t^i as

$$p(s_t \mid z_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(s_t - s_t^i) \tag{18}$$

where δ denotes the Dirac function. It can be shown that the approximation (18) approaches the true posterior density as $N \rightarrow \infty$. Moreover, one can now obtain an empirical estimate of the state s_t via maximum likelihood or through a different statistical measure.

Importance sampling causes most particles to have negligible weight after a few iterations, which is called the *sampling degeneracy* problem. To avoid this, one can apply a re-sampling scheme, which can generally be done by replicating particles in proportion to their weights. Sequential importance resampling (SIR) is a very commonly used particle filtering algorithm. This

process eliminates samples with low weights and chooses better particles [90]. The graphical representation of SIR particle filters [37] is shown in Figure 3. On the other hand, it produces the loss of diversity for a set of particles, i.e., particles with high weights are selected too much, and thus the others disappear as time goes on. Therefore, all of the particles will eventually collapse to the same value. To alleviate this *sample impoverishment*, Some techniques have been proposed to improve the sample diversity in the literature, such as regularized particle filter [75] and the Markov chain Monte Carlo (MCMC) move step [36]; see [90] for detailed discussion.

CHAPTER III

OBJECT TRACKING AND TARGET REACQUISITION BASED ON 3D RANGE DATA FOR MOVING VEHICLES

In this chapter, we propose a reliable algorithm to track a deformable object in a time-varying sequence of 3D range data by combining particle filtering and geometric active contours. In addition, this work will present an on-line shape learning method based on principal component analysis (PCA) to reacquire track for an object of interest in the event that it disappears from the field of view and reappears later.

We separate the object tracking problem into two parts based on the idea of *deformotion* [121] in which the overall motion of an object can be described by the combination of a global motion and local deformations: the prediction of temporal change of the object's position and the object segmentation for each frame. More specifically, the proposed algorithm is comprised of the filtering process (for tracking the global motion obtained by particle filtering) and the segmentation process (for tracking local deformations achieved by active contours).

As part of the segmentation approach, a weighted depth map is defined, and the curve evolution for object extraction is performed via the Bhattacharyya gradient flow which is exploited due to its robustness to noise of a cluttered depth map. Range information is filtered by the proposed weighting scheme to improve the quality of active contour segmentation, and to better estimate the global motion of the object. In the filtering process, particle filtering and active contours are employed in conjunction to estimate the global position of the object in the weighted image space. For motion estimation, comparable approaches using the filtering schemes in conjunction with active contour models may be found in [87, 88, 24, 97]. In addition, we define a certain similarity shape energy based on a statistical shape model to achieve continuous tracking without the prior shape model for a uncertain period of time even if the object is not observable, i.e., the object being tracked is fully occluded or leaves the scene completely.

Lastly, we propose a dynamic weighting scheme in which the weighted depth maps are dynamically generated based on the previous tracking results. We further assume that the motion of the object is smooth throughout a sequence so that the objects, segmented via active contours, between consecutive frames overlap each other. Based on this assumption, the global motion of the tracked object can be estimated by active contour segmentation using the dynamically weighted depth maps without particle filtering. This technique using the dynamic weighted depth map provides the computational efficiency and simplification for the tracking framework.

The remainder of this chapter is organized as follows. In the next section, we provide some related works to the proposed approach. In Section 3.2, we give an overview some fundamental theories used in the proposed algorithms such as geometric active contours driven by the Bhattacharyya gradient flow and statistical shape representation. Section 3.3 describes the proposed algorithms for object tracking in stereo image sequences. First, a segmentation method that incorporates depth information is presented. Next, a filtering algorithm used to estimate global motion is discussed. In Section 3.4, the proposed method of disappearance and reappearance handling is presented and discussed. Experimental results on a real image sequence using the tracking framework presented in Section 3.5 are shown in Section 3.6. Section 3.7 introduces a method to dynamically generate weighted depth maps. In Section 3.9, a tracking framework based on the dynamic weighting scheme presented in Section 3.8 was tested on challenging real-world sequences. Lastly, we conclude this chapter and discuss possible future research directions in Section 3.10. Much of this chapter is based on [59, 57].

3.1 Introduction and Related Work

Tracking moving objects is an important task in computer vision. The task of tracking is to locate and segment the object of interest from the background at each frame of the image sequence. Numerous algorithms have been proposed to segment and track the given object in recent years (e.g., see [9, 8, 42, 123] and references therein).

Target reacquisition has been extensively studied in visual surveillance systems to maintain the identity of a moving object across cameras with overlapping [11] as well as non-overlapping [43] fields of view. Much of the work adopted a probabilistic approach to compute the correspondence

matching of objects between multi-cameras [11, 43]. One usually integrates spatio-temporal relationships with appearance information of a desired target. The work in [16] constructs a graph based track initialization for target matching, and the Kalman filter is utilized to predict the movement of a target in the blind region. In [14], a learning method based on the prior knowledge of camera network topology is employed to build a spatio-temporal link, and a normalized histogram is used for the appearance model. In comparison to a number of approaches for visual surveillance systems, we address the reacquisition problem in the context of object tracking with a moving camera. More specifically, the proposed method uses a single stereo camera to capture a moving vehicle in an uncontrolled environment. Thus, the spatio-temporal link between cameras is not applicable in our framework. In addition, we take advantage of shape information obtained from the proposed contour-based tracker to continuously track the target instead of a histogram based appearance model many times adopted in the surveillance system.

First of all, we briefly revisit several attempts to specifically handle occlusions in the context of visual tracking [19, 76, 124]. Such occlusions can occur when another object lies between the target and a camera, or the target occludes parts of itself. In general, most methods incorporate shape information of an object of interest into a tracking framework online [124] or offline [125] to make up for poor distinguishable statistics between the object and background or missing parts of the object. To this end, a shape prior can be obtained or learnt from linear principal component analysis (PCA) if the assumption of small variations in shape holds [63]. Otherwise, for highly deformable objects, locally linear embedding (LLE) [86] or nonlinear PCA [19] may be employed. Yilmaz *et al.* [124] present a contour-based method for tracking non-rigid objects using a Bayesian framework in which the contour is implemented via level set methods and a shape energy term is added to handle occlusions. In [87, 88], Rathi *et al.* propose a particle filtering algorithm, which incorporates geometric active contours for tracking deformable objects. They also embed shape information into the update step of particle filtering to cope with occlusions.

The works mentioned so far have only considered partial occlusion handling with prior knowledge based on a shape model. Severe or complete occlusions have also been explored in the following papers. In [76], Nguyen and Smeulders present a template matching method using an appearance model smoothed temporally by the Kalman filter. Here, they define a maximal bound on the length

of recoverable occlusions. After a certain number of frames, the temporal filter slowly incorporates the occlusion into the object model. Similarly, in [56] a minimum cost computation-based template selection is introduced to handle occlusions assuming the object reappears within a certain time and spatial region. However, both algorithms fail track continuously when the object does not appear again within pre-defined time or when the object presents an appearance not accounted for in the template. Jackson *et al.* [45] propose an explicit model for the motion of contours that is enforced by higher-order motion models, such as inertia. In [5], Bartesaghi and Sapiro propose a tracking algorithm using spatio-temporal minimal surfaces in 3D space-time domain to handle severe or total occlusions of an object.

For segmentation and tracking, we utilize 3D range data computed by determining correspondences and disparity maps from stereoscopic image sequences. Depth information from disparity maps has been used in the literature to improve the performance of object tracking and segmentation tasks. For the detection of moving objects, Talukder and Matthies [110] introduced the 3D measured flow field by combining the 2D optical flow and the disparity flow from depth maps. Also, the authors in [3] presented a segmentation method using edge maps for a stereo image as an external energy for active contours. The edge combination image is introduced in [34], which is obtained from both the original intensity image and stereo-driven depth maps for contour-based segmentation. Both approaches improve the capability in image segmentation but they have difficulties in dealing with weak edge levels or boundary gaps. Adaptive object segmentation methods based on depth and spatio-temporal information are presented in [115]. Here, different disparity planes are used to segment the area with smooth disparity variations. In [94], a depth-based tracker is introduced to enhance object tracking using time-of-flight range imaging sensor in the limited illumination. The combination method of stereo-derived edge information and image intensity is proposed in [71] for contour-based segmentation. In this chapter, we combined the advantages of geometric active contours and range information for better performance of object segmentation and tracking in a stereo sequence.

3.2 Background

In this section, we briefly introduce geometric active contours driven by the Bhattacharyya gradient flow and PCA-based shape representations.

3.2.1 Geometric Active Contours Driven by the Bhattacharyya Gradient Flow

Many active contour segmentation energies have been proposed to drive the segmenting curve toward the boundaries of an object [12, 13, 20]. In the present work, we incorporate an active contour model driven by maximizing a statistical measure of dissimilarity (known as the Bhattacharyya distance) between the curve's interior and exterior. We employ this particular model to deal with highly cluttered depth maps because it is robust against noisy images and initial curve placement while retaining the ability to segment multi-modal objects in cluttered scenes. Furthermore, it has been shown to be computationally efficient enough for use in tracking applications [33, 73].

We now offer a brief summary of the Bhattacharyya distance, and its use in active contour segmentation. From Section 2.1 of chapter 2, in the level set methods, a closed curve C is represented as the zero level set of a higher dimensional function ϕ, which is typically chosen to be a signed distance function: $\phi < 0$ inside C and $\phi > 0$ outside C. Now, the curve can be described by an implicit surface: $C = \{\mathbf{x} \mid \phi(\mathbf{x}) \equiv 0,\ \mathbf{x} \in \Omega\}$, where Ω is the domain of an image $I(\mathbf{x})$: $\mathbb{R}^2 \rightarrow \mathbb{Z}$ which is mapping to the photometric variable $z \in \mathbb{Z}$. The Bhattacharyya distance between two probability density functions (pdfs) is defined by $D_B = -\log(B)$ where

$$B = \int_{\mathbb{Z}} \sqrt{P_i(z)P_o(z)}dz, \tag{19}$$

which is the Bhattacharyya coefficient and varies between 0 and 1 (0 indicates a complete mismatch while 1 represents perfect similarity). P_i and P_o are pdfs defined inside and outside curve C, respectively:

$$P_i(z) = \frac{\int_\Omega K(z - I(\mathbf{x}))H(-\phi(\mathbf{x}))d\mathbf{x}}{\int_\Omega H(-\phi(\mathbf{x}))d\mathbf{x}}, \quad P_o(z) = \frac{\int_\Omega K(z - I(\mathbf{x}))H(\phi(\mathbf{x}))d\mathbf{x}}{\int_\Omega H(\phi(\mathbf{x}))d\mathbf{x}} \tag{20}$$

where K is the given kernel. Popular choices for the kernel K are either Gaussian function or the Dirac delta function. $H(\cdot)$ is the Heaviside step function such that $H(\phi) = 1$ for $\phi \geq 0$ and $H(\phi) = 0$ for otherwise.

Now, the optimal level set function with a regularizing term for smooth curve evolution is defined as:

$$\phi^{\star} = arg \inf_{\phi}\{E_{image}\} \tag{21}$$

where

$$E_{image} = B(\phi) + \alpha \int_{\Omega} \|\nabla H(\phi)\| \, d\mathbf{x} \tag{22}$$

where α is a user defined regularization constant ($\alpha > 0$) and ∇ denotes the gradient. After differentiating P_i and P_o and derivative of B with respect to ϕ, we have the gradient flow for the level set evolution as follows (see Appendix B for details):

$$\frac{\partial \phi}{\partial t} = \delta(\phi)(\alpha\kappa - S)$$

where

$$S = \frac{B}{2}\left(\frac{1}{A_i} - \frac{1}{A_o}\right) + \frac{1}{2}\int_Z K(z - I(x)) \left(\frac{1}{A_o}\sqrt{\frac{P_i(z)}{P_o(z)}} - \frac{1}{A_i}\sqrt{\frac{P_o(z)}{P_i(z)}}\right) dz. \tag{23}$$

Here $\delta(\cdot)$ is the delta function. A_i and A_o are the areas inside and outside the segmenting curve, respectively. The curvature κ is given by: $\kappa = \text{div}\{\frac{\nabla\phi}{\|\nabla\phi\|}\}$. The gradient flow of (23) will converge to a contour which maximizes the discrepancy between the distributions inside and outside the curve [73].

3.2.2 Shape Representation

Shape statistics are necessary to represent the modes and variations of shapes. In this work, we employed the eigen-shape based representation using PCA, introduced by [63]. The basic idea is that a given shape may be represented by a linear combination of a set of eigen-shapes after projecting the shape into the eigen-space. Shapes are represented by the signed distance function. In the level set framework, ϕ_i is the signed distance function determined by the zero level set, corresponding to the segmenting curve C_i. The mean shape μ for n shapes is computed as $\mu = \frac{1}{n}\sum \phi_i$. The mean-offset, $\phi_i - \mu$, is placed as a column vector in an matrix M ($r \times n$) where $\phi_i \in \mathbb{R}^r$ and r is the number of dimensions. The covariance matrix in shape, $\frac{1}{n}MM^T$, is decomposed by the singular value decomposition (SVD):

$$U\Sigma U^T = \frac{1}{n}MM^T \tag{24}$$

where U is a matrix whose column vectors represent the set of orthogonal modes of shape variations, and Σ is a diagonal matrix of corresponding singular values. The estimated shape for ϕ is given by:

$$\tilde{\phi} = U_k U_k^T (\phi - \mu) + \mu \tag{25}$$

where U_k is a matrix consisting of the first k columns of U.

3.3 Visual Tracking using 3D Information

3.3.1 Segmentation using 3D Range Data

There are many algorithms used to find stereo correspondence from a pair of rectified stereo images (see [72, 51, 41, 98] and references therein). The output of these algorithms is a disparity map showing the relative displacement of a pixel from one image to the other. In the proposed algorithm we compute disparity maps using a naive sum of squared difference stereo matching scheme that yields much lower-quality matching results than available algorithms. This is done to show the robustness of proposed algorithm against poor-quality stereo reconstruction. Better results could be expected if more accurate stereo correspondences were used.

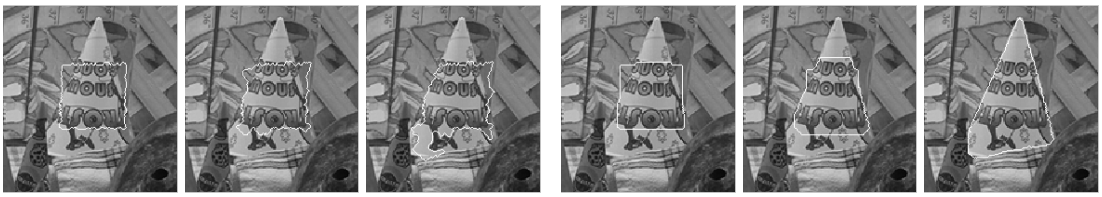
The disparity space is defined as a three-dimensional projective transformation of 3D space $[x,\ y,\ z]^T$ for a given pair of rectified stereo images [98]. The correspondence between a pixel $(x,\ y)$ in a reference image and a pixel $(x',\ y')$ in a matching image is given by

$$x' = x + D(x,\ y), \quad y' = y \tag{26}$$

where $D(x,\ y)$ is a uni-valued disparity function (or map) with respect to a reference image. This disparity map can yield a 3D range value (or a depth value) for each pixel in the scene if the cameras' calibration information is known [40]. The depth map $D_p(\mathbf{x})$: $\mathbb{R}^2 \rightarrow \mathbb{D}$, which maps the depth value $d \in \mathbb{D}$, of a pair of images is represented in terms of disparity function $D(\mathbf{x})$. Thus, given the position vector $[x,\ y,\ z]^T$ of an object in the three-dimensional image space, z can be substituted with the depth value d: $[x,\ y,\ d]^T$.

Depth information allows simplification and improvement of the segmentation task on a single image as well as sequence of images for the following reasons. First, the object's 3D range values will be quasi-homogenous, even though intensities inside the object have different photometric intensities. Furthermore, the size of the object being tracked can be approximated with the associated

(a)



(b) (c)

Figure 4: (a) Original left image (left) and a disparity map (right). Segmenting the cone: initial, intermediate and final contours using (b) only image intensity information, and (c) a disparity map. Images are zoomed in for better visualization. The test data set is taken from [99].

depth values. These capabilities allow us to estimate parameters for active contour models such as the maximum iteration number for curve evolution or the size of initial contours. Thus, if intensity distribution of an object is a multi-modal and non-homogenous, the segmentation using 3D range data can be a useful scheme to handle the following problems:

- Extracting the outlines of an object in a highly textured image.
- Object segmentation in scenes where the foreground and background have a significant depth difference (e.g., moving vehicles, such as an aircraft in the sky or a car on a road).

Some results of segmentation using active contours driven by the Bhattacharyya gradient flow with and without using the disparity map are shown in Figure 4. The cone of interest is located in the rear-center of the given image. In Figure 4(b), the cone is a highly textured object, thus, the segmenting curve does not enclose the whole outline of the cone. Instead, it converged to the edges inside the cone. When using the disparity information, an acceptable segmentation is achieved as shown in Figure 4(c).

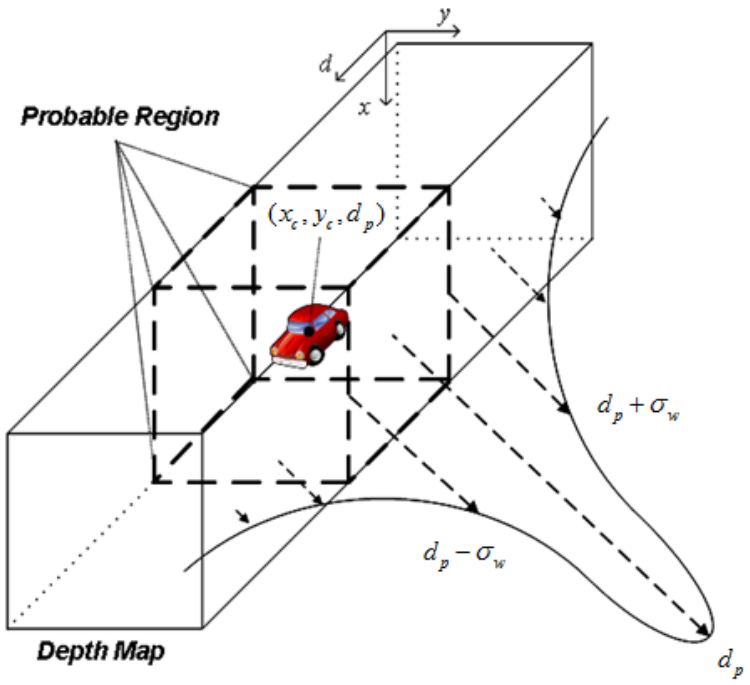


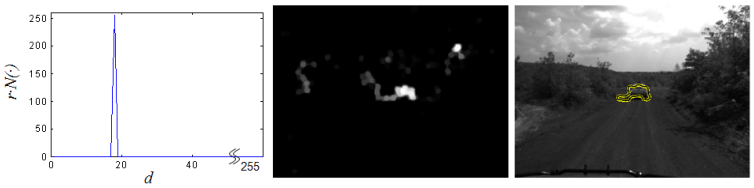
Figure 5: The concept of the weighting range information with a Gaussian Kernel $\mathcal{N}(d_p,\ \sigma_w^2)$. The dashed cube describes the probable region, which is weighted heavily by the given kernel.

3.3.2 Weighted Depth Maps

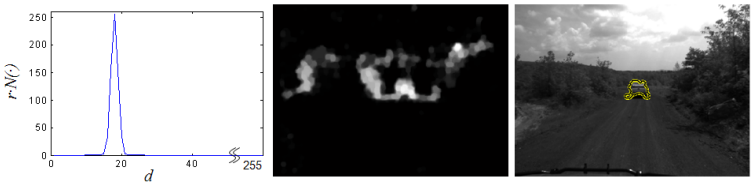
The basic idea of the proposed segmentation scheme is to weight depth values in proportion to the probability of the appearance of the object of interest prior to curve evolution. By applying a Gaussian weighting filter to the range data, unimportant noise and far-away structures are suppressed or eliminated. This method allows for improved segmentation results and faster convergence. In addition, it restricts the image information according to the depth value of the object and weights the most probable regions highly.

A kernel f_w is a non-negative real-valued integrable weighting function. For example, the Gaussian function or a uniform function may be used. Specifically, we chose a Gaussian function whose mode is the certain depth value d_p and its variance is σ_w^2: $\mathcal{N}(d_p,\ \sigma_w^2)$. The new weighted depth map is given by

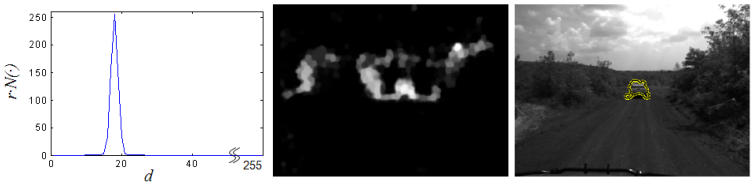
$$\begin{aligned} D_w(\mathbf{x}) &= f_w(D_p(\mathbf{x})) = \gamma \cdot \mathcal{N}(d_p,\ \sigma_w^2) \otimes f_s(\cdot) \\ &= \frac{\gamma}{\sqrt{2\pi}} \exp\left\{ \frac{(D_p(\mathbf{x}) - d_p)^2}{2\sigma_w^2} \right\} \otimes f_s(\cdot) \end{aligned} \tag{27}$$



(a)



(b)



(c)

Figure 6: Gaussian weighting functions (left column), the weighted depth maps (middle column), and segmentation results (right column). The Gaussian weighting kernels are constructed with (a) $\sigma_w^2 = 0.1^2$, (b) $\sigma_w^2 = 1^2$, and (c) $\sigma_w^2 = 10^2$. γ and d_p are 255 and 18, respectively.

where γ is a weight parameter ($\gamma > 0$) and $f_s(\cdot)$ is a smoothness regularization filter. The weight parameter γ is assigned as 255 in our case, but it can be any positive value without loss of generality. The typical choice for $f_s(\cdot)$ is a morphological smoothing filter or an isotropic Gaussian filter. The size of the filter depends on the noise present in the sequence. Figure 5 shows the concept of the proposed weighting on range data, where the probable region is defined as the limited or sliced area within one standard deviation of the given depth value d_p.

The variance of the kernel f_w is chosen based on how confidently one can estimate the depth of the target. A large variance should be chosen if the image data is noisy or has poor contrast. Alternatively, if the variance is very small, the target will appear more prominent, but loss of the object information can occur. In addition, this value should be adjusted online according to the depth and size of the object. Figure 6 shows the segmentation results using Gaussian function with different variances. Note that good results are obtained when σ_w^2 is chosen so that the weighting function accurately reflects the variation of depth values seen in the target. In practice, we have

(a) (b)

Figure 7: (Upper row of (a) and (b), left-to-right) Original left image with an initial contour, depth map, and the weighted depth map with (a) $f_w \sim \mathcal{N}(18,\ 1^2)$, and (b) $f_w \sim \mathcal{N}(57,\ 2.5^2)$. (Lower row of (a) and (b), left-to-right) Segmentation results using only image intensity information, depth map, and the weighted depth map. Images are zoomed in for better visualization. $\gamma = 255$.

(a) Occlusion handling using range information

(b) Segmentation of a car through partial occlusions

Figure 8: (a) Original depth map (left), a result after weighting range data (middle), and a result after a morphological smoothing filter (right). (b) Original left image with an initial contour (far left), segmentation results using only image intensities (the second column), depth map (the third column), and the weighted depth map (far right).

found that $\sigma_w^2 = [1^2,\ 3^2]$ works well in most cases.

In this work, segmentation is achieved with the active contours driven by the Bhattacharyya flow introduced in Section 3.2.1. The kernel K in (23) is defined by delta function $\delta(\cdot)$. By substituting the weighted depth map $D_w(x,\ y)$ in (27) for an image space $I(\mathbf{x})$, we have the speed term S for (23) as

$$S = \frac{B}{2}\left(\frac{1}{A_i} - \frac{1}{A_o}\right) + \frac{1}{2}\int_{\mathbb{Z}} \delta(z - D_w(\mathbf{x}))\left(\frac{1}{A_o}\sqrt{\frac{P_i(z)}{P_o(z)}} - \frac{1}{A_i}\sqrt{\frac{P_o(z)}{P_i(z)}}\right) dz. \qquad (28)$$

Figure 7 shows a robust segmentation result in a highly cluttered environment. As can be seen in the results, since the backgrounds (e.g., the ground, skies and trees) are far from the trucks, the proposed algorithm is well suited for this type of segmentation. Note that the range information in

Figure 7 is scattered and the trucks are not easily distinguished from the backgrounds. However, the proposed weighting scheme allows the trucks to be segmented without loss of information and prevents leaks into nearby structures as shown in Figure 7. Moreover, it provides an essential cue in dealing with partial occlusions. As seen in Figure 8, the stop sign can be ignored by the proposed weighted depth maps. Here, the stop sign disappears after weighting the depth maps and the car is highlighted after applying a smoothing filter $f_s(\cdot)$, and the segmentation results via active contours in Figure 8 shows that the contours leak into the stop sign and nearby structures excepts for the case of using the proposed scheme.

3.3.3 Filtering for Motion Estimation

We assume that the location of an object is within the probable region defined in Section 3.3.2, and the difference of the object's location between consecutive frames is small: $[x \pm \delta x,\ y \pm \delta y,\ d \pm \delta d]^T$ given the position vector of the object is $[x,\ y,\ d]^T$ in three-dimensional image space, where δx, δy, and δd are the small perturbations of each coordinate. The global motion of an object is described by the transition of the coordinates of its centroid and depth value $[x_c, y_c, d_p]^T$, and the local motion is represented by the segmenting curve C evolved by the gradient flow in (23) and (28). Thus, the state vector is given by

$$s(t) = \begin{pmatrix} \rho \\ d_p \\ C \end{pmatrix}(t) \tag{29}$$

where $\rho = (x_c,\ y_c)^T$ is the centroid of an object and C denotes the contour coordinates represented by the zero level set of ϕ. When a new image pair arrives, we have an observation $z(t)$ which is the available weighted depth map $D_w(t)$ at time t: $z(t) = D_w(t)$. Let a prior density be the proposal distribution to simplify the dynamic model and let resampling be applied at every time step [90, 106]. We then have the weight update equation from (16) as

$$w_t^i = p(z_t \mid s_t^i)$$

where the likelihood of the observation is given by

$$p(z_t \mid s_t^i) = \exp\{-E_{image}(s_t^i,\ z_t)\}. \tag{30}$$

The proposed estimation scheme using particle filtering is described as follows:

1) Prediction Step:

   (a) Generate the N particles, $\{s_t^i\}_{i=1\ldots N}$ around s_{t-1} in the following manner:

$$\rho_t^i = \rho_{t-1}^i + u_t, \quad u_t \sim \mathcal{N}(0,\ \sigma_u^2)$$
$$(d_p)_t^i = (D_p)_{t-1}(\rho_t^i)$$
$$C_t^i = C_{t-1} + \rho_t^i.$$

2) Update Step:

   (a) Make the weighted depth map based on $(d_p)_t^i$ from (27):

$$(D_w)_t = \ (f_w)_t((D_p)_t)$$

   where

$$(f_w)_t \sim \mathcal{N}(\ \frac{1}{N}\sum_{i=1}^{N}(d_p)_t^i,\ \sigma_w^2\ ).$$

   (b) Evolve the curve over the weighted depth map for each s_t^i for l iterations. l is generally a small number, which is carefully selected in consideration of the degree of trust of the system and measurement models to avoid *sample degeneracy* and *sample impoverishment*. In our proposed filtering framework, the careful choice of the number of curve evolutions, introduced in [88, 97], is adopted to solve both problems. More specifically, if l is chosen to be too large, this can lead to *sample degeneracy* due to the loss of temporal coherency. Likewise, choosing l to be too small results in *sample impoverishment* because all particles would not move to the region of the high likelihood [88, 97]. $l = 3$ is selected experimentally for robust results of our experiments in Section 3.6.

   (c) Compute the importance weights using (30):

$$\tilde{w}_t^i = \exp\{-E_{image}(\ (D_w)_t,\ s_t^i\ )\}$$

   and normalize:

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{i=1}^{N}\tilde{w}_t^i}.$$

(d) The posterior distribution of the system is represented by a set of weighted particles as

$$p(s_t \mid z_{1:t}) = \sum_{i=1}^{N} w_t^i \delta(s_t^i).$$

(e) Resample N particles according to $p(s_t \mid z_{1:t})$ by using the generic resampling scheme introduced in [90, 106]: $\{s_t^i,\ 1/N\}_{i=1}^N$.

(f) The best fitting curve, which maximizes the dissimilarity between two distributions for inside and outside the curve in (19) and minimizes the segmentation energy in (22), is selected for the measurement model.

(g) The centroid of the selected curve is taken as the centroid of the system and its depth value is given by

$$(d_p)_t = \frac{\int_\Omega (D_p)_t(\mathbf{x}) H(-\phi_t(\mathbf{x}))d\mathbf{x}}{\int_\Omega H(-\phi_t(\mathbf{x}))d\mathbf{x}}.$$

3.4 Target Reacquisition

3.4.1 Disappearance and Reappearance Handling

In this section, we address the estimation problem of continuing to track an object of interest even when it is partially or fully occluded during the course of tracking. This naturally leads to the questions of how to detect the disappearance of an object and how to recognize the reappearance of an object and to reacquire track.

To solve these problems, we analyze the characteristics of disappearance and reappearance in dynamic imagery. For example, the size of an object will decrease as it disappears and increase as it reappears. Thus, disappearance detection may be achieved by comparing the average size of the tracked object from past frames with the object's current size. Reappearance detection may be accomplished by finding an object that is most similar to the tracked object before it was lost. Some assumptions about the location at which the object will reappear are also necessary to avoid the computational complexity of searching the entire domain and increase the possibility of reacquisition by reducing the probability of unexpected detection of a noise. In this work, we assume that the object being tracked will appear near the last known position of the object (e.g., if the object disappears on left side of the image, it will also reappear on the left side). That is, the object's centroid when it reappears, ρ_r, is constrained to be near the position of its centroid before disappearance,

ρ_d. Also, the probable region when the object reappears is assumed to be the last probable region before its disappearance. Thus, the searching region for reappearance detection is described by

$$\begin{aligned} \rho_r &= [\rho_d - \varphi,\ \rho_d + \varphi] \\ (d_p)_r &= (d_p)_d \end{aligned} \tag{31}$$

where φ is the longest value between the average lengths of the major and minor axes of the object, and $(d_p)_d$ and $(d_p)_r$ are the depth value when the object disappeared and will reappear, respectively. Note that these assumptions may restrict the applicability of the proposed algorithms for certain very important applications such as video surveillance in which it would be quite difficult to estimate reappearance using the preceding set-up. Thus, in this work, the proposed algorithms are tested on appropriate scenarios for these assumptions, such as tracking moving vehicles. Indeed, the proposed algorithms show the best performance in following a moving car on a road with various cluttered backgrounds as shown in our experiments in Section 3.6.

Shape analysis based on multiple feature correspondences is an indispensable tool to cope with disappearance and reappearance. The contour tracker, proposed in the previous sections, provides useful feature information of the segmented object in terms of the level set function ϕ, such as outline, position, size, curvature, and aspect ratio of the object. From this information, we propose the following shape energy-based disappearance and reappearance handling scheme. The basic idea of the proposed algorithm is to define the shape similarity energy based on the feature information of past frames, and then detect the object's return by comparing this feature information with the current frame and finding a candidate region that closely matches the template shape. To create the template shape, we assume that the object being tracked deforms gradually from frame to frame, and its shape remains similar while out of view. If all sets of shapes lie on a linear manifold and their deformations are small, PCA can be used to provide a shape representation [86]. Thus, the template shape of an object is obtained using PCA-based shape statistics described in Section 3.2.2.

The template shape is based on the history of the segmented object over previous frames up to the current, rather than learned priors. More specifically, the shape of an object at time step t is represented as a binary mask selecting regions inside the closed curve C_t. Thus, we can write this using the Heaviside function as $H(-\phi_t)$. Finally, the template shape at time step t is defined as the

mean shape over all previous frames

$$\phi_{\mathrm{TS}} = \frac{1}{n-1}\sum_{t=1}^{n-1} H(-\phi_t),\ n > 1. \tag{32}$$

If the new shape, ϕ_{new}, of the same class of the object is received online at time step t, the shape dissimilarity function for ϕ_{new} can be obtained from (25) and (32) as

$$\Gamma_1^2(\phi_{\mathrm{new}},\ \phi_{\mathrm{TS}}) = \frac{\left\|\log(1+|{U_k}^T(H(-\phi_{\mathrm{new}})-\phi_{\mathrm{TS}})|)\right\|}{1+\left\|\log(1+|{U_k}^T(H(-\phi_{\mathrm{new}})-\phi_{\mathrm{TS}})|)\right\|} \tag{33}$$

where $|\cdot|$ and $\|\cdot\|$ denote the absolute value and the Euclidean norm, respectively.

The size and curvature differences between ϕ_{new} and ϕ_{TS} are considered as a feature distinction function described by

$$\Gamma_2^2(\phi_{\mathrm{new}},\ \phi_{\mathrm{TS}}) = \left[\log\frac{\int_\Omega H(-\phi_{\mathrm{new}}(\mathbf{x}))d\mathbf{x}}{\int_\Omega H(-\phi_{\mathrm{TS}}(\mathbf{x}))d\mathbf{x}}\right]^2 + \left[\kappa(H(-\phi_{\mathrm{new}})) - \kappa(\phi_{\mathrm{TS}})\right]^2. \tag{34}$$

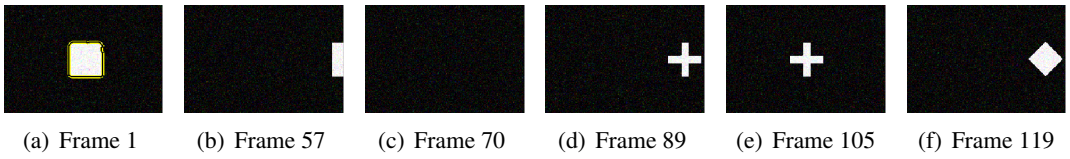
Other shape descriptors can be added to regulate the feature distinction, such as the aspect ratio or the degree of symmetry. Here, the aspect ratio of ϕ is defined as the ratio between the lengths of the major and the minor axes, where these lengths are obtained from eigen-values of the covariance matrix of ϕ. Also, the symmetry degree can be estimated by computing a difference between the original shape and its reflected superimposition. One can find several relevant shape descriptors in [21]. Now, the similarity shape energy is obtained from:

$$E_{shape} = \exp\left\{-\frac{1}{2}(\Gamma_1^2+\Gamma_2^2)\right\}. \tag{35}$$

From (35), disappearance occurs when the shape similarity energy is sufficiently small. Indeed, E_{shape} will typically decrease gradually as the object disappears from the scene. Reacquisition of the tracked object is completed by evolving the curve in the search region until a candidate with sufficiently high shape similarity energy is detected. Thus, we determine disappearance or reappearance if E_{shape} in (35) satisfies certain criteria. To detect disappearance, we have

$$E_{shape}(\Gamma_1^2,\ \Gamma_2^2,\ (\phi_{\mathrm{TS}})_{t-1},\ \phi_t|\ z_{1:t}) < \eta_d < \overline{E_{shape}} \tag{36}$$

and, to detect reappearance, the condition is given by



(a) Frame 1 (b) Frame 57 (c) Frame 70 (d) Frame 89 (e) Frame 105 (f) Frame 119

(g) Frame 135 (h) Frame 149 (i) Frame 165 (j) Frame 176 (k) Frame 180 (l) Frame 195

Figure 9: Shape Sequence. Frame order: from left to right, top to bottom. Note that the reacquisition of the tracked square is achieved after (k) frame 180 , and that the other shapes except the tracked square are ignored during the course of tracking.

(a) ϕ_{TS} (b) 0.66 (c) 0.62 (d) 0.5 (e) 0.37 (f) 0.52 (g) 0.45 (h) 0.39 (i) 0.21

Figure 10: (a) Template shape. From (b) to (e) Shape energies corresponding to various shapes of the shape sequence. From (f) to (i) Shape energies of other shapes.

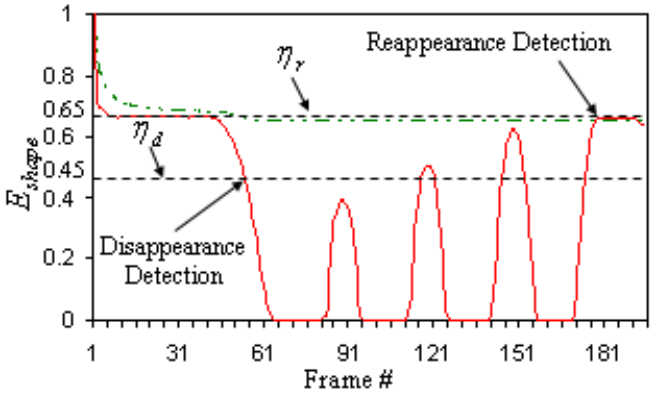


Figure 11: Graph of the shape energy for each frame of the shape sequence. The dashed-dotted line and solid line denote $\overline{E_{shape}}$ and E_{shape}, respectively. $\eta_d = 0.45$ and $\eta_r = 0.65$.

$$E_{shape}(\Gamma_1^2,\ \Gamma_2^2,\ (\phi_{\mathrm{TS}})_{t_d-1},\ \phi_t|\ z_{t_d:t}) > \eta_r > \overline{E_{shape}} \tag{37}$$

where $\overline{E_{shape}}$ is the arithmetic mean of E_{shape} up to $t-1$. η_d and η_r are positive thresholds between 0 and 1 for each case of detection. The value t_d is the time when the object disappears. If the condition (36) is *true* during tracking, then one considers target as having disappeared, and similarly for the reappearance condition (37). Note that the proposed approach is performed without

the prior training sets and puts no time limitation on the object's absence.

In Figure 9, a synthetic image sequence is used to test the algorithms for disappearance and reappearance handling based on the proposed similarity shape energy in (35). The sequence includes several different shapes and is generated with Gaussian white noise of zero mean and 0.01 variance. Figure 10 shows the template shape as well as the shape energies corresponding to each shape of the shape sequence. Some different shapes' energies are also given in Figure 10 to show the variation of the shape energy with respect to the template shape. The change of $\overline{E_{shape}}$ and E_{shape} for this sequence is shown in Figure 11. Note that none of the shape energies satisfied the detection condition in (37) except the square. Thus, only the square is acquired again during the course of tracking.

3.4.2 Discussion of the Reacquisition Method

Measuring the similarity between two objects is an important task in many areas, such as image retrieval, object identification and visual tracking. A comprehensive review of such similarity measures is beyond the scope of this work; see [31] and references therein. Accordingly, in this subsection, the shape similarity based reacquisition method proposed in this work is discussed and compared with other similarity measures in the context of visual tracking.

It is quite common in object tracking that an image region that includes the object of interest is represented by a template model of the pixel intensities or other relevant feature information, and then is compared to candidate regions to determine the displacement of the object over consecutive frames. To compare or evaluate similarities between the template and an observed image (or a region), the Bhattacharyya distance [17, 113], Kullback-Leibler divergence [32], and the normalized mutual information [66] have been proposed to evaluate the degree of a similarity between intensity distributions. These probability-distance measures usually use histogram based distributions, which describe the image region of interest. However, such histogram based approaches are insensitive to the possible deformations of an object due to the lack of spatial information. Taking into consideration the invariance to translation and robustness to lighting changes, some edge information and the orientation histogram may be employed to measure the similarity with respect to the Euclidean distance as in [118, 67]. In addition, Birchfield and Rangarajan [7] introduced a spatial histogram,

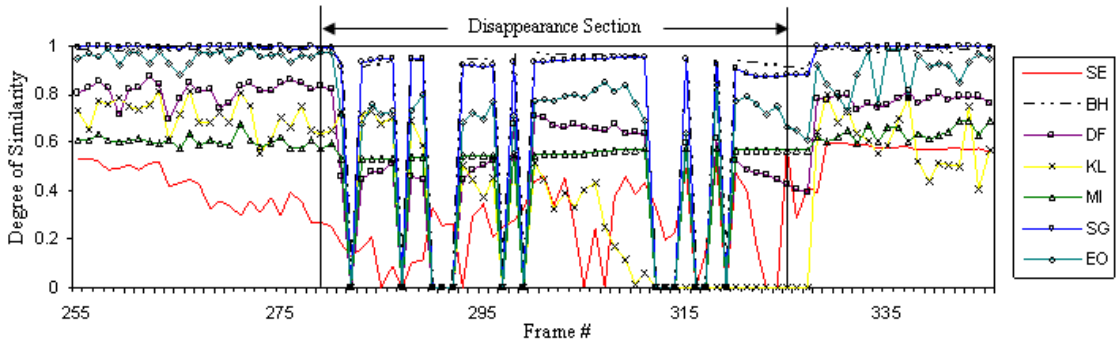


Figure 12: Graph of the degree of similarity for some similarity measures over some frames of the sequence in Figure 14; the proposed shape-similarity energy (SE), the Bhattacharyya distance (BH), the diffusion distance (DF), Kullback-Leibler divergence (KL), the normalized mutual information (MI), the spatiogram (SG), and the edge-orientation histogram (EO) were tested. In the degree of similarity, 0 and 1 indicate complete mismatch and perfect similarity, respectively.

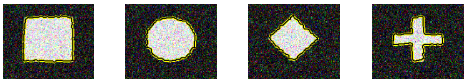
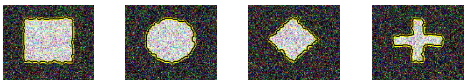
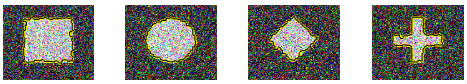
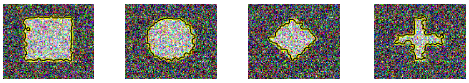
Table 1: Table displaying statistical information in the disappearance section during ≈50 frames of the sequence in Figure 12. MAX, MIN, AVG, and VAR denote a maximum value, a minimum value, an average, and a variance, respectively.

	SE	BH	DF	KL	MI	SG	EO
MAX	0.601	0.983	0.833	0.796	0.621	0.997	0.975
MIN	0	0	0	0	0	0	0
AVG	0.257	0.708	0.438	0.268	0.420	0.699	0.576
VAR	0.028	0.171	0.077	0.083	0.060	0.167	0.118

or *spatiogram*, in which some information of the geometry of an object feature distribution is involved. This work showed improved performance of tracking for a mean-shift based tracker. On the other hand, the similarity measure used in [7] is not suited for small spatial changes of the given object's features. To overcome this, the work of [18] formulates a similarity measure based on the spatiogram and the Bhattacharyya coefficient for robust object localization. Recently, Ling and Okada [65] exploited the diffusion process to derive a cross-bin histogram distance, called *diffusion distance*, which is robust to the object's deformation and a lighting change in histogram-based local descriptors.

We have tested the similarity measures discussed above in the real sequence of Figure 14 of the experimental section 3.6 to comparatively evaluate the proposed shape-similarity energy. Figure 12

Table 2: Quantitative results for the robustness of the proposed shape energy to a noise for various shapes of the sequence in Figure 9. Shape energies corresponding to various shapes in the diverse noise levels are displayed at the bottom of each image. Gaussian noises with $\sigma_n^2 = 10\%$ (first row), $\sigma_n^2 = 25\%$ (second row), $\sigma_n^2 = 50\%$ (third row), and $\sigma_n^2 = 100\%$ (fourth row) were added, respectively. Refer to Figure 10 for the template shape and the shape energy of the Gaussian noise with $\sigma_n^2 = 1\%$. $d_{\sigma_n^2}$ denotes the difference of shape energies between $\sigma_n^2 = 1\%$ and $\sigma_n^2 = 100\%$.

σ_n^2	Shapes with a noise and their energies			
10%		0.637	0.567	0.421	0.321
25%		0.63	0.567	0.425	0.309
50%		0.621	0.565	0.415	0.299
100%		0.615	0.555	0.392	0.268
$d_{\sigma_n^2}$		0.045	0.065	0.108	0.102

shows the graph of the degree of similarity for similarity measures, such as the Bhattacharyya distance, the diffusion distance, Kullback-Leibler divergence, the normalized mutual information, the spatiogram, and the edge-orientation histogram. In Figure 12, we see that most measures moderately catch the similarities between objects over the consecutive frames in the presence of the object but their similarity values are severely unstable during the disappearance portion of the video. To reacquire the tracked target, a similarity measure should provide the discriminating threshold to separate the state of the target's presence from the state of the target's absence. To evaluate this, statistical information, such as a maximum value and a variance of similarity values during the disappearance section of the tested sequence are computed and recorded in Table 1. From the results of Table 1, other measures' variances and maximum values in the disappearance section are too large

to allow the distinguishable detection of the target's reappearance. However, the proposed shape-similarity energy gradually decreases while the target is turning to the left or right and its variance and maximum value in the disappearance section are of a size such that the detection threshold can be properly selected in the detection conditions in (36) and (37).

From the comparative study with other similarity measures, the present shape-based similarity measure seems to be more discriminative and shows the applicability for target reacquisition of a moving vehicle. However, since the tolerance of the template shape depends on the variation of the shape learned online from previously tracked object, the proposed method can fail to reacquire the track of the object due to deformations and shapes that have not been learned in our scheme. Theoretically, one can increase the number of prior training sets to increase the possibility of reacquisition of the track, but it does not always guarantee reacquisition of the object with an unregistered shape. Thus, as mentioned in the previous section 3.4.1, it must be pointed out that it is necessary to assume that the tracked object maintains its shape similarly while it has been out of view regardless of the number of training shapes before disappearance.

Finally, the robustness of the proposed shape energy to noise was tested as seen in Table 2. The shapes tested were generated with diverse noise levels of Gaussian noise whose variance ranges from $\sigma_n^2 = 1\%$ to $\sigma_n^2 = 100\%$. Note that even though all similarity-shape energies of the shapes decrease as the noise level increases, the rectangular shape has the biggest shape energy and its energy difference between noises (or decreasing rate) is the smallest because it is the most similar shape to the template.

3.5 Tracking Framework I

Tracking algorithms are integrated in a unified framework. The diagram of the entire tracking procedure is illustrated in Figure 13. The various algorithms are shown in the order in which they are performed, and the loop indicates iteration to the next frame. In our applications, depth maps of pairs of stereo images are constructed from a stereo matching system at each time step. The initialization process to identify the position of the object of interest is only performed for the first frame. The diagram shows that the procedure starts with global and local estimation of the object, and finishes with shape analysis for disappearance and reappearance handling.

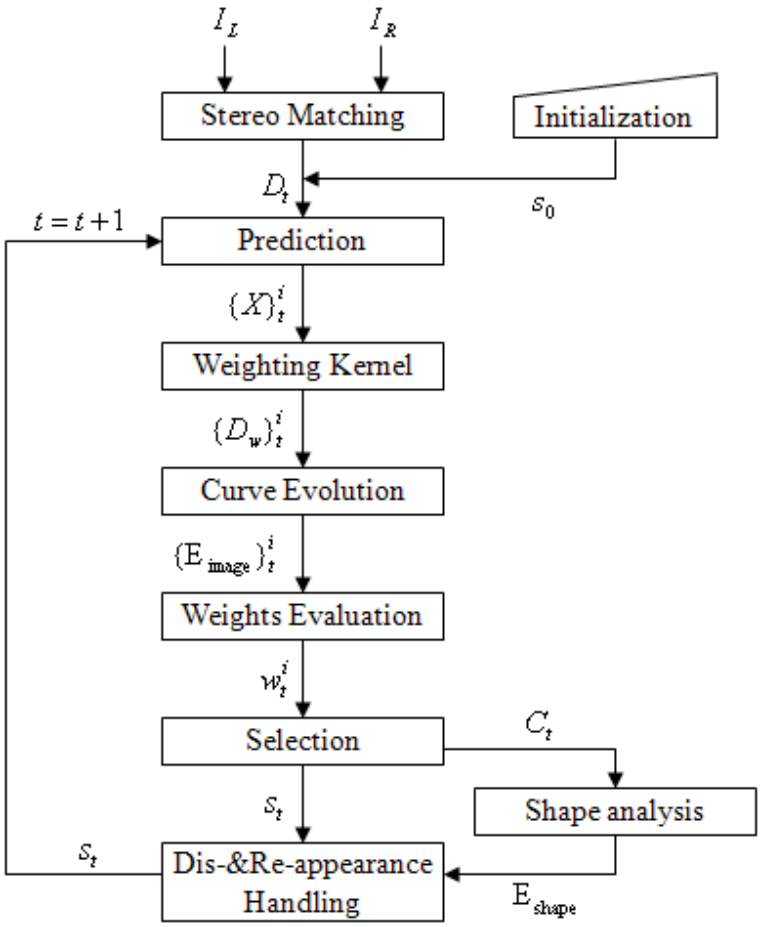
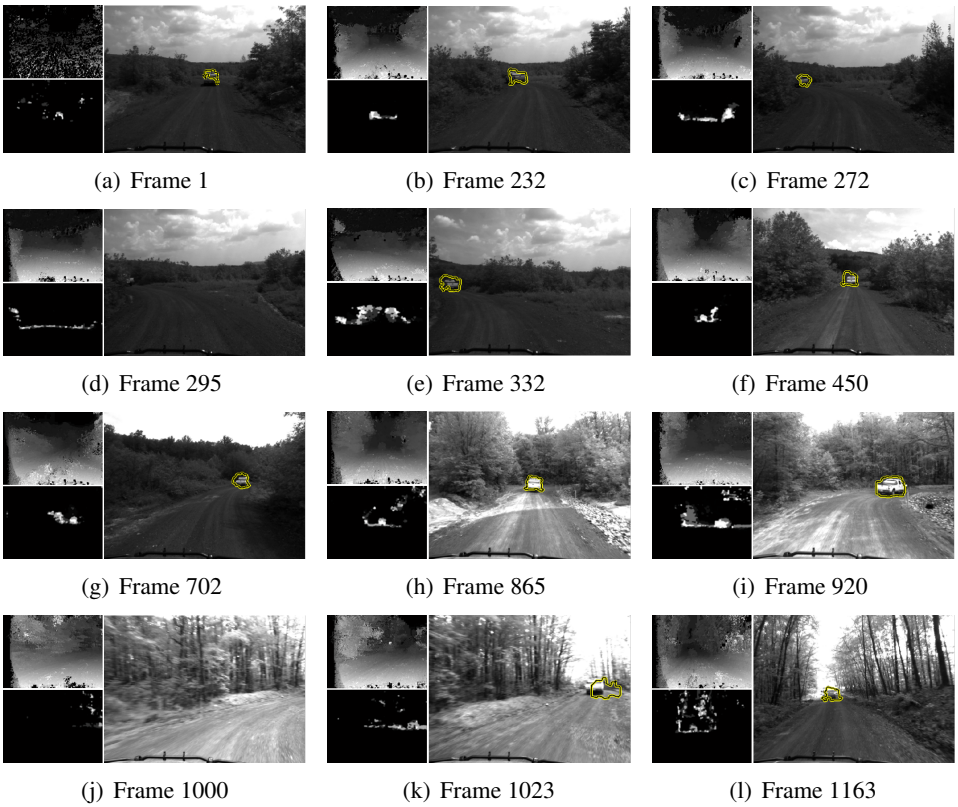


Figure 13: The overall framework diagram of the proposed algorithms (I_L and I_R denote a left image and a right image, respectively).

3.6 Experiments I

The proposed tracking framework in Section 3.5 was tested on three different videos of moving vehicles obtained from stereoscopic image sequences. While the experimental sequences are composed of numerous stereo image pairs, results are shown only on the left image for simplicity. Additionally, the computed depth images from all sequences have low contrast, high noise, and cluttered backgrounds. In the experiments most of the parameters were held fixed across all trials. Specifically, the number of particles $N = 40$ was chosen empirically to give good coverage without adding significant computational burden. Overall, the tracker produced accurate track signals on the three sequences with an acceptable computation time of approximately 20 seconds per frame on a 3.6GHz Windows machine with 2GB of RAM. The value of σ_w^2, which is used to control the



(a) Frame 1 (b) Frame 232 (c) Frame 272

(d) Frame 295 (e) Frame 332 (f) Frame 450

(g) Frame 702 (h) Frame 865 (i) Frame 920

(j) Frame 1000 (k) Frame 1023 (l) Frame 1163

Figure 14: Truck Sequence I. Frame order: from left to right, top to bottom. The upper and bottom left images to each frame are a depth map and the weighted depth map, respectively. Note that disappearance and reappearance handling is achieved between (c) and (e), and between (i) and (k). The condition of illumination is changed after 865th frame.

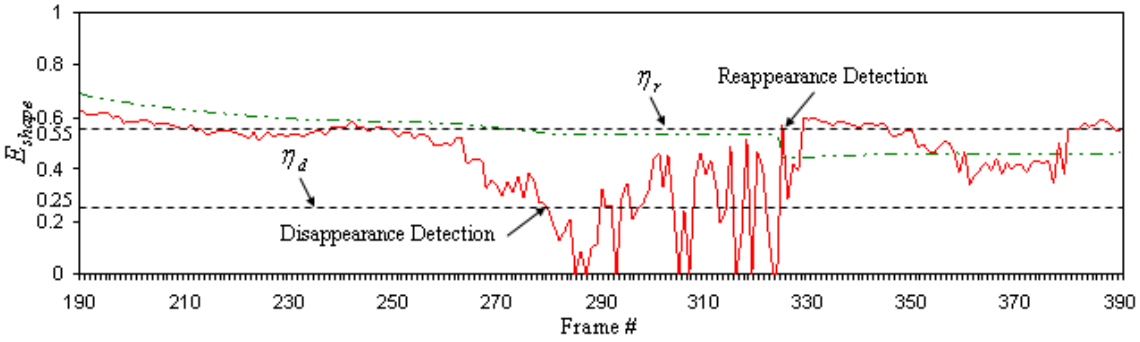


Figure 15: Graph of the shape energy for some frames of the truck sequence I. The dash-dot line and solid line denote $\overline{E_{shape}}$ and E_{shape}, respectively. Note that the detection of disappearance and reappearance of the tracked truck is achieved with $\eta_d = 0.25$ and $\eta_r = 0.55$.

Gaussian weighting on depth information, is set automatically in our experiments as described in Section 3.3.2.

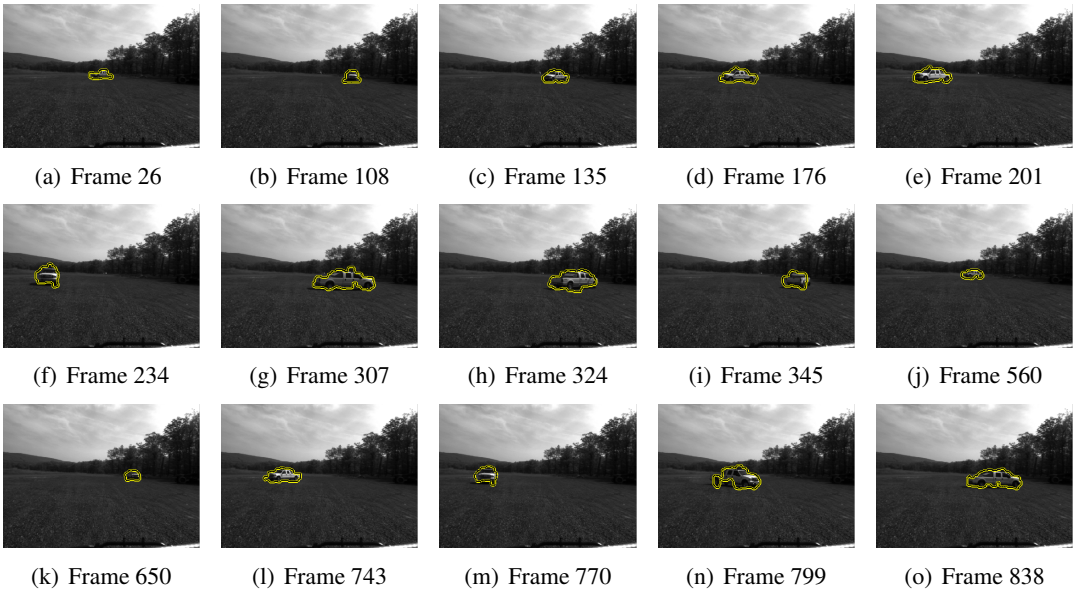
To deal with disappearance and reappearance of the object, two thresholds η_d and η_r were selected based on the expected shape variation within a sequence and observations of how this affects tracking performance. For example, η_d should not be too small or the object may not be identified as "disappeared" before it disappears from the view. Similarly, the object may not be robustly reacquired if η_r is too large. A detailed analysis of these parameters is given in the sections below and in Figures 15 and 18. However, for our experiments, choosing parameters in the ranges $\eta_d = [0.2, 0.3]$ and $\eta_r = [0.5, 0.6]$ gave robust detection.

3.6.1 Truck Sequence I

This sequence is taken from a moving camera following a truck while trying to keep a constant distance of approximately 20 meters. It is comprised of 1168 frames in which the truck moves in and out of the field of view several times: once while turning left (frames 278 to 328), and once while turning right (frames 950 to 1020). In addition, the sequence shows changing illumination conditions and significant clutter. Weighted depth maps of each frame are obtained using the Gaussian function with $\sigma_w^2 = [1^2,\ 1.2^2]$ in (27). Tracking results with the proposed algorithms are shown in Figure 14. We can see from the results that even though the truck has very weak edges and the background is quite cluttered, the truck is robustly tracked. Furthermore, the results demonstrate the capability of the proposed scheme to handle disappearance and reappearance. For disappearance and reappearance detection, η_d and η_r are selected as 0.25 and 0.55, respectively. The change of $\overline{E_{shape}}$ and E_{shape} during frames 190 through 390 is shown in Figure 15. The shape energy of the truck decreases as the truck turns to the left and eventually drops below η_d as it disappears from the image domain. After ≈50 frames, the truck's reappearance is detected when it becomes visible and its shape-similarity energy satisfies the reappearance condition in (37).

3.6.2 Truck Sequence II

In this sequence, the truck moves randomly in clockwise and counter-clockwise loops. Note that the shape of the truck deforms significantly whenever it turns left or right. In addition, 3D range values (i.e., depth values) decrease and increase markedly as it moves back and forth. The sequence



(a) Frame 26 (b) Frame 108 (c) Frame 135 (d) Frame 176 (e) Frame 201

(f) Frame 234 (g) Frame 307 (h) Frame 324 (i) Frame 345 (j) Frame 560

(k) Frame 650 (l) Frame 743 (m) Frame 770 (n) Frame 799 (o) Frame 838

Figure 16: Truck Sequence II. Frame order: from left to right, top to bottom.

includes 866 frames with a cluttered background. The parameter σ_w^2 is chosen to be $[1^2,\ 3^2]$. It has a larger range than in other experiments because the change of the object's size is much more dramatic. Figure 16 shows several frames that demonstrate the robust tracking results. Note that no disappearance or reappearance of the truck took place because it was visible throughout the sequence.

3.6.3 Van Sequence

Figure 17 shows the proposed algorithm robustly tracking a van in a cluttered urban environment throughout a 45 frame sequence. Initially, the van takes a turn to the left causing significant shape deformation. Then it gradually moves away from the camera until it vanishes into the distance. Weighted depth maps are created with $\sigma_w^2 = [1^2,\ 2^2]$. The disappearance of the van is successfully detected with $\eta_d = 0.2$. This smaller value is chosen because of the drastic change in perceived shape energy of the target (side of the van to back of the van) in just 10 frames. The change of $\overline{E_{shape}}$ and E_{shape} for this sequence is shown in Figure 18. The shape energy of the van decreases gradually and eventually drops below the detection threshold η_d as it disappears into the distance.

(a) Frame 1 (b) Frame 3 (c) Frame 16 (d) Frame 35 (e) Frame 45

Figure 17: Van Sequence. Frame order: from left to right. Note that the detection of disappearance is achieved in the last part of the sequence.

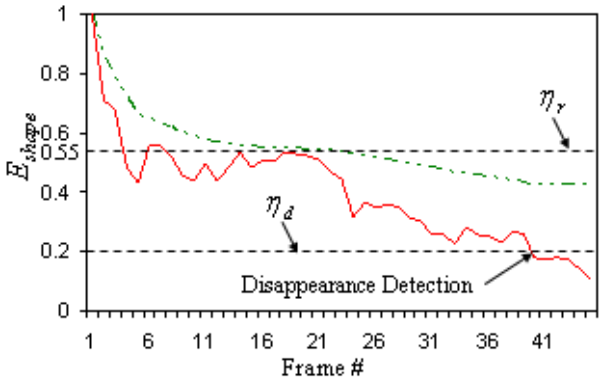


Figure 18: Graph of the shape energy for each frame of the van sequence. The dash-dot line and solid line denote $\overline{E_{shape}}$ and E_{shape}, respectively. $\eta_d = 0.2$ and $\eta_r = 0.55$.

3.7 Dynamic Weighting Scheme

In this section, we introduce the extension of the method presented in previous sections. Compared with the approaches in Section 3.3, the weighted depth map is now dynamically generated according to the previous results of tracking and range information. Here we assume that the object of interest shows complete smooth motion with no abrupt changes. From this assumption, the proposed dynamic weighted depth maps are used to improve segmentation via active contours as well as to estimate global motion of an object instead of the use of particle filters. In addition, the convex hull of the segmented contour of a previous frame is used for an initial contour of a current frame. These new approaches lead to better computational efficiency and simplification of the entire tracking framework presented in Section 3.5.

The global position of an object is described by its centroid and depth value $[x_c, y_c, d_p]^T$, and the local motion is represented by the segmenting curve C evolved by the gradient flow in (28). In this work, the problem of tracking the position of an object at time t is estimating the appropriate

parameters, $(d_p)_t$ and $(\sigma_w^2)_t$, for the weighted depth map in (27). The depth value is obtained from the segmented curve of the previous frame:

$$(d_p)_t = \frac{\int_\Omega (D_p)_{t-1}(\mathbf{x}) H(-\phi_{t-1}(\mathbf{x}))d\mathbf{x}}{\int_\Omega H(-\phi_{t-1}(\mathbf{x}))d\mathbf{x}}. \tag{38}$$

The variance, σ_w^2, should be chosen to reflect the variation of depth values inside the tracked object. Thus, it is defined by the combination of mean and variance of depth values inside segmented contour of the previous frame:

$$(\sigma_w^2)_t = \frac{1}{2}\left(\frac{\zeta - (d_p)_t}{\max(I_t) - \min(I_t)} + V\right) \tag{39}$$

where

$$V = \frac{\int_\Omega [(D_p)_{t-1}(\mathbf{x}) H(-\phi_{t-1}(\mathbf{x})) - (d_p)_t]^2 d\mathbf{x}}{\int_\Omega H(-\phi_{t-1}(\mathbf{x}))d\mathbf{x}}, \tag{40}$$

and ζ is a maximum allowance for σ_w^2. In our work, ζ was chosen to be about 20. The centroid of the object is taken as the centroid of the segmented curve C_{t-1} and an initial contour for C_t is obtained from the convex hull of C_{t-1}.

3.8 Tracking Framework II

The proposed tracking framework is composed of two parts: the dynamic weighting scheme to estimate the parameters for the weighting kernel f_w to achieve the weighted depth maps, and active contour segmentation for tracking deformations of an object. The diagram of the entire tracking procedure is illustrated in Figure 19. The initialization process is carried out manually to identify the position of the target at the first frame. Note that the proposed tracking framework is not based on a filtering scheme. Thus, it does not exploit the dynamics underlying object's motion (i.e., without a prediction model), and it depends only on segmentation (i.e., with a measurement model) to estimate the state of the object of interest. This approach improves the speed of tracking algorithm because global motion is estimated by the proposed dynamic weighted depth maps without particle filters, which is a sampling-based method with high computational complexity. Note that this is done by the assumption that the difference of the target's location between consecutive frames is small so that the objects, segmented via active contours, between the previous frame and the current frame overlap each other.

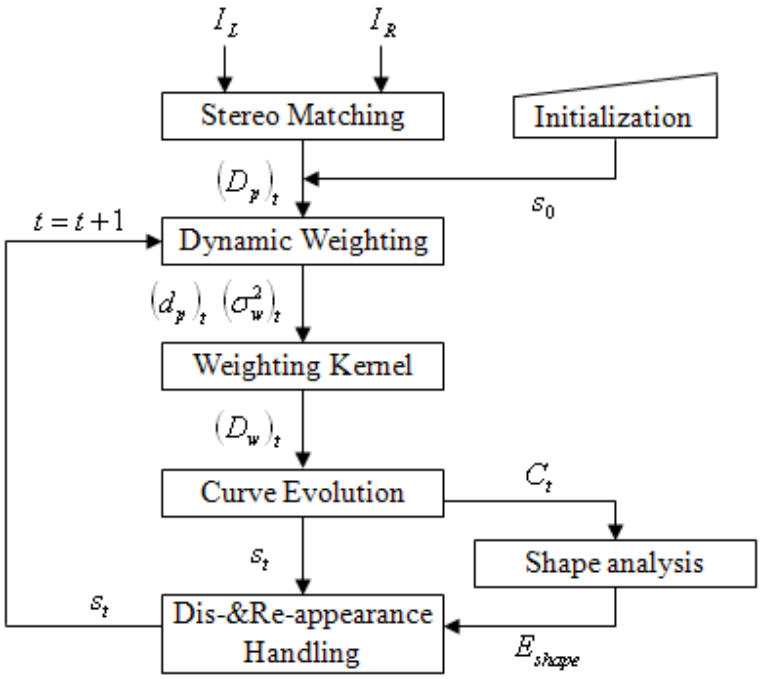


Figure 19: The diagram of the proposed tracking framework using the dynamic weighting scheme. I_L and I_R denote a left image and a right image, respectively.

3.9 Experiments II

The proposed tracking framework in Section 3.8 was tested on several illustrative sequences in real environments. The experimental results are shown only on the left image of stereo image pairs. Compared to the experimental results of Section 3.6, the weighting parameters are dynamically changed at each frame according to the segmented curve and range information of the previous frame. In addition, the results are obtained with approximately 5 seconds per frame on a 3.6GHz and 2GB Windows machines. It shows less computational time than the tracking framework in Section 3.5.

3.9.1 Truck Sequence III

In this sequence, a stereo camera is fixed and the trucks move in clockwise loops in Figure 20 (a) and in counter-clockwise loops in Figure 20 (b). Thus, the trucks' shapes change whenever they move. In particular, the truck in Figure 20 (b) shows significant changes of its shape and size. In addition, the condition of illumination is severely changed between frames due to moving clouds in the sky. However, the trucks are accurately tracked by the proposed algorithms despite the large

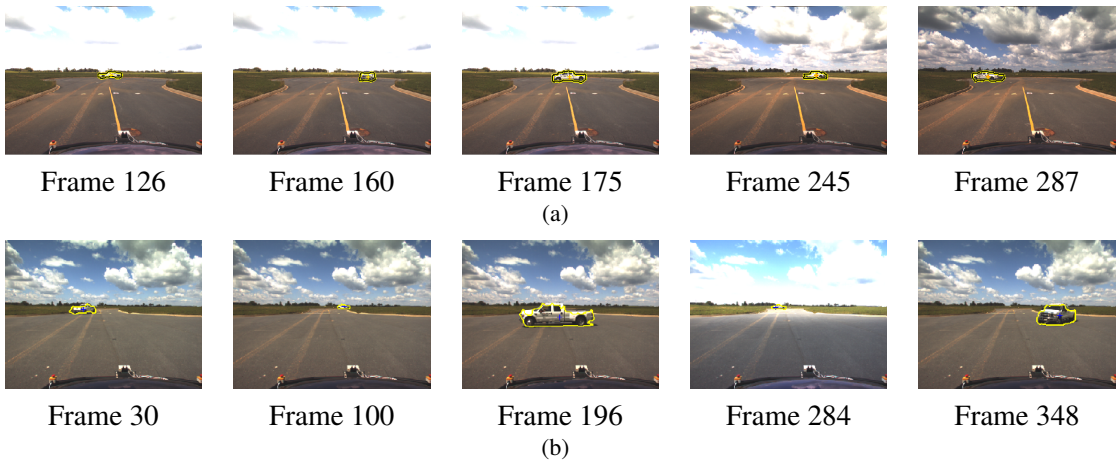


Figure 20: Truck Sequence III. Frame order: from left to right. Note that the condition of illumination is severely changed between Frame 175 and Frame 245 in (a), and Frame 196 and Frame 348 in (b).

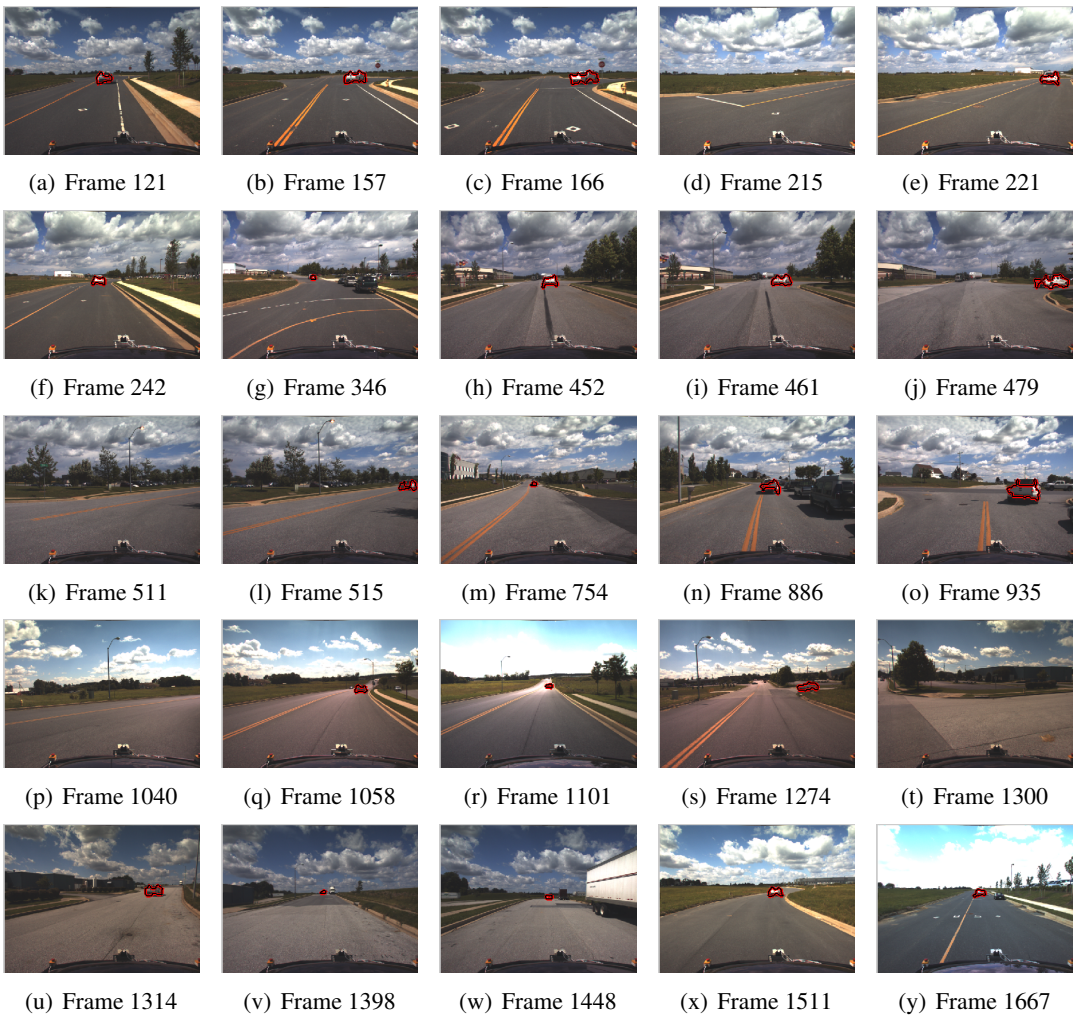
shape and illumination variations as shown in Figure 20.

3.9.2 Truck Sequence IV

This sequence is taken from a moving camera following a truck in a cluttered urban environment, which includes 1667 frames. In this sequence, the truck moves in and out of the field of view several times whenever it turns left or right. However, as shown in Figure 21, the truck is robustly tracked and its track is maintained throughout the sequence. In addition the truck is successfully reacquired whenever it appears again by the proposed reacquisition methods with $\eta_d = 0.3$ and $\eta_r = 0.45$. In Figure 22, the tracker introduced in Section 3.3 eventually lost the track due to a nearby truck showing a similar depth value. However, the tracker using the dynamic weighting scheme introduced in Section 3.7 still maintained the track without the divergence of curve evolution.

3.10 Chapter Conclusion

In this chapter, we described a straightforward approach for tracking moving objects using 3D range data by incorporating the contour-based tracker in conjunction with PCA-based shape analysis. Region-based active contours are employed to track deformations of an object using the Bhattacharyya gradient flow on the Gaussian weighted depth map. A particle filtering scheme combined with the active contour model is utilized for global position estimation of the target. In addition,



(a) Frame 121 (b) Frame 157 (c) Frame 166 (d) Frame 215 (e) Frame 221

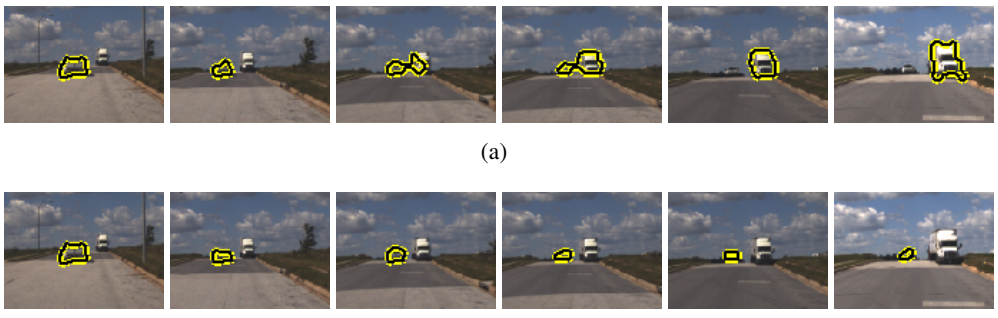
(f) Frame 242 (g) Frame 346 (h) Frame 452 (i) Frame 461 (j) Frame 479

(k) Frame 511 (l) Frame 515 (m) Frame 754 (n) Frame 886 (o) Frame 935

(p) Frame 1040 (q) Frame 1058 (r) Frame 1101 (s) Frame 1274 (t) Frame 1300

(u) Frame 1314 (v) Frame 1398 (w) Frame 1448 (x) Frame 1511 (y) Frame 1667

Figure 21: Truck Sequence IV. Frame order: from left to right, top to bottom. Note that disappearance and reappearance handling is achieved between (c) and (e), and between (j) and (l), and between (o) and (q), and between (s) and (u). The condition of illumination is significantly changed at several frames.



(a)

(b)

Figure 22: The tracking results (a) without and (b) with the proposed dynamic weighting scheme. Frame order: from left to right. (a) Loss of tracking via the method introduced in Section 3.3 due to a nearby truck with a similar depth value to the truck being tracked. (b) The track is maintained by the method using the dynamic weighting scheme introduced in Section 3.7.

we proposed similarity shape energy in order to handle the possible disappearances and reappearances of the tracked object. We demonstrated the ability of the proposed method to track targets in cluttered environments and to detect automatically the target after it has gone out of frame and then reappeared. Moreover, the dynamic weighting scheme was also proposed to effectively use the tracking result of the previous frame in generating the weighted depth maps. This approach simplified the tracking framework and reduced the computational complexity by rejecting the use of a sampling-based prediction step.

The proposed algorithm has some limitations which we intend to overcome in our future work. First, the proposed algorithm is limited in that it will fail if the object reappears with an unexpected shape that is unlike those seen previously. Second, the proposed algorithm will lose the target if the unexpected object occludes the target or appears in a similar depth position with a similar shape to the tracked object in the course of tracking. It is a possible solution for some limitations to incorporate the information of color and shape of an object from an original sequence into the proposed algorithms. Also, our future research will include the challenging problem of tracking multiple objects with varying shapes. To improve reliability and accuracy of tracking, more sophisticated shape analysis will be needed.

CHAPTER IV

2D-3D VISUAL POSE TRACKING AND OCCLUSION HANDLING USING THE 3D MODEL OF A RIGID OBJECT

In this chapter, we address the problem of 2D-3D pose estimation. Specifically, we propose an approach to jointly track a rigid object in a 2D image sequence and to estimate its pose (position and orientation) in 3D space, relative to a calibrated camera. The relative pose can be described in terms of trajectories on a transformation matrix [68]. Thus, the goal of this work is to find a 3D transformation matrix to optimally estimate the pose of an object of interest.

To this end, first, we propose the Monte Carlo based sampling method to estimate a 3D transformation matrix for 2D object tracking and 3D pose estimation. In addition, we take advantage of knowledge of a 3D model of an object. This allows the tracker to capture various aspects (or quasi-deformation) of an object with respect to its dynamic motion, while the methods based on learning a collection of 2D shape priors have difficulties in completely describing them. The 3D models[1] used in this chapter consist of an elephant, a car, and a helicopter as shown in Figure 23.

Next, in order to effectively handle occlusions and improve the performance of 2D-3D pose tracking, we propose a new tracking framework based on filtering. Specifically, we employ the more natural particle filtering scheme to generate and propagate the translation and rotation parameters in a decoupled manner. In particular, we revisit a joint 2D segmentation/3D pose estimation technique of [25], and then extend the framework by incorporating a particle filter to track the object and its corresponding pose. Moreover, to allow the algorithm to continuously track the object in the presence of occlusions, an occlusion detection and handling scheme is developed as follows. First, a histogram-based appearance model is created and updated to detect occlusion. Second, a dynamical choice of how to invoke the objective functional is performed online to handle the occlusion. The decision how to control the degree of dependencies between predictions and measurements of the system is based on the degree of occlusion and the variation of the object's pose.

[1] All 3D models for rigid objects were acquired from the method presented by Yezzi and Soatto [122, 119].

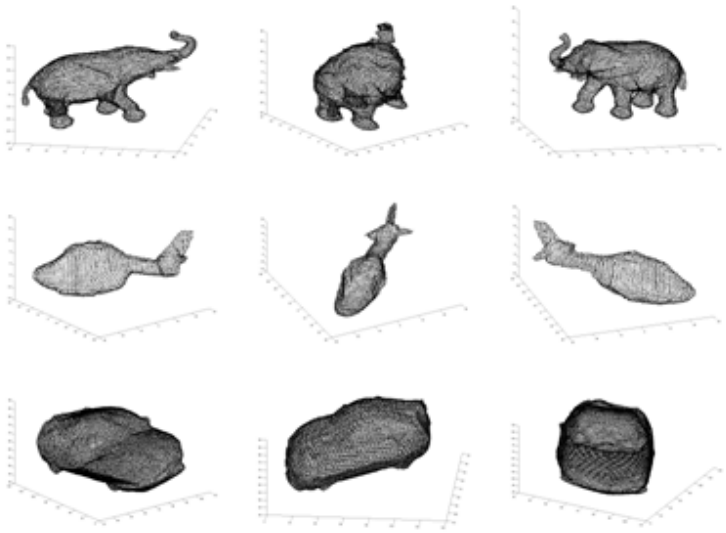


Figure 23: Different views of 3D point models used in this chapter. From top row to bottom row: an elephant, a helicopter, and a car.

The remainder of this chapter is organized as follows. In the next section, we provide some related work to the proposed approach. In Section 4.2, we define notation and terminology used in this chapter. Section 4.3 presents the first approach for 2D-3D visual pose tracking, which is based on Monte Carlo Sampling on SE(3) and 3D shape knowledge of an object. In Section 4.5, the first tracking framework presented in Section 4.4 is tested on a real image sequence. Next, the second approach for 2D-3D pose estimation based on particle filtering is proposed in Section 4.6. This section includes the proposed novel occlusion handling scheme for the task of object tracking. In Section 4.8, we provide experiments on both synthetic and real life imagery in hopes of highlighting the viability (and limitations) of the proposed algorithm, which is summarized in the tracking framework presented in Section 4.7, in the context of visual tracking. Lastly, we conclude this chapter and discuss possible future research directions in Section 4.9. Much of this chapter is based on [60, 61].

4.1 Introduction and Related Work

Visual tracking has been a significant topic of research in the field of computer vision; see [9, 42, 104, 123] and references therein. The ultimate goal of visual tracking is to continuously identify the 3D location of an object of interest from an image sequence. This amounts to what is known as

the 2D-3D pose tracking problem [68, 40]. However, due to the difficulty of developing a tractable solution for estimating the 3D position from a 2D scene, many researchers have tacitly restricted the tracking problem to be concerned with only the relative 2D location of the object in which segmentation is often employed in conjunction with Kalman or particle filters [48, 37]. Recent techniques attempt to revisit the 2D-3D pose tracking problem for challenging scenarios by leveraging on 2D image segmentation to estimate the 3D location [54, 89, 92, 100]. While impressive results have been obtained that rival both pose tracking and segmentation based algorithms, these schemes did not fully exploit the underlying system dynamics that is inherent in any visual tracking task. Thus, in order to effectively treat the temporal nature of the observed 2D scene, we propose to extend a similar framework proposed by [25] in which we now incorporate a particle filter to perform 3D pose estimation of a rigid object.

Several algorithms have been introduced to solve the 2D-3D pose tracking task. In general, they are based on local or region attributes for feature matching. For example, such features include points [1], lines [27], polyhedral shape [70], complete contours [93, 30], or surfaces [100]. Specifically, in [70], the authors perform a 2D global affine transformation as an initialization to 3D pose estimation, and then the 3D object pose is computed by an energy minimization process with respect to an approximate polyhedral model of the object. The authors in [30] present a 3D pose estimation algorithm by using visible edges. That is, they use binary space partition trees for finding and determining the visible line to track the edges of the model. However, since these methods rely on local features, the resulting solutions may yield unsatisfactory results in the presence of noise or cluttered environments. To overcome this, an early attempt to couple segmentation and pose estimation is given in [89]. In their work, the authors propose a region-based active contour that employs a unique shape prior, which is represented by a generalized cone based on a single reference view of an object. More recently, authors in [25] as well as Schmaltz *et al.* [100] propose a region-based model scheme for 2D-3D pose tracking by projecting a 3D object onto a 2D image plane such that the optimal 3D pose transformation coincides with the correct 2D segmentation. In a similar fashion, Kohli *et al.* [54] proposed a joint segmentation and pose estimation scheme using the graph cut methodology. Although these methods perform exceptionally well for many cases, they do not exploit the underlying dynamics inherent in a typical visual tracking task. We should

note, in the context of the proposed work, the incorporation of system dynamics can be viewed as an extension to these baseline algorithms. To the best of our knowledge, this is the first attempt to exploit the temporal coherency of a video sequence for these particular class of algorithms.

In addition to the aforementioned approaches for 2D-3D pose tracking, many works may be found in the literature which purely focuses on restricting the visual tracking problem to the 2D domain. Because a complete overview of existing methods is beyond the scope of this chapter, we just consider those methods that employ various filtering schemes such as the Kalman filter [112], unscented Kalman filter [48, 114], and particle filter [37, 29] as well as explicit algorithms for occlusion handling [19, 125]. Specifically, the authors in [64, 15] employ a finite dimensional parameterization of curves, namely B-splines, in conjunction with the unscented Kalman filter for rigid object tracking. Generalizing the Kalman filter approach, the work in [118] presents an object tracking algorithm based on particle filtering with quasi-random sampling. Since these approaches only track the finite dimensional group parameters, they cannot handle local deformations of the object. As a result, several tracking schemes have been developed to account for deformation of the object via the level set technique [78, 102]. In relation to our work, some early attempts for 2D visual tracking using level set methods can be found in [80, 121]. In particular, authors in [121] propose a definition of motion for a deformable object. This is done by decoupling an object's motion into a finite group motion known as "deformotion" with that of deformation, which is any departure from rigidity. Building on this, authors in [88] introduce a deformable tracking algorithm that utilizes the particle filtering framework in conjunction with geometric active contours. Other approaches closely related to these frameworks are given in [112, 83, 84]. Here the authors use a Kalman filter for predicting possible movements of the object, while the active contours are employed only for tracking deformations of the corresponding object.

In addition to employing filtering schemes to increase the robustness of tracking, many algorithms invoke a systematic approach to handle occlusions. We should note that although the main contribution of our work focuses on employing particle filtering to estimate the 3D pose during 2D visual tracking, a neat feature of the resulting methodology is its ability to handle occlusions effectively. Thus, we briefly revisit several attempts to specifically handle occlusions in the context of

visual tracking [19, 76, 124]. Such occlusions can occur when another object lies between the target and a camera, or the target occludes parts of itself. In general, most methods incorporate shape information of an object of interest into a tracking framework online [124] or offline [125] to make up for poor distinguishable statistics between the object and background or missing parts of the object. To this end, a shape prior can be obtained or learnt from linear principal component analysis (PCA) if the assumption of small variations in shape holds [63]. Otherwise, for highly deformable objects, locally linear embedding (LLE) [86] or nonlinear PCA [19] may be employed. Like that of [25] as well as other 2D-3D pose tracking algorithms, we assume we have prior knowledge of the 3D shape of interest for which we would like to estimate the corresponding 3D pose from the 2D scene. Occlusion handling is then accomplished by adjusting the "trust" between the prediction and measurement model according to the degree of occlusion and variation of the object's 3D pose from previous accumulated results.

4.2 Notation and Terminology

Let $\mathbf{X} = [X, Y, Z]^T$ be the coordinates of a point in $\mathbb{R}^3$ with respect to the referential of the camera. Here, it is assumed that the calibrated camera is already given and is modeled as a pinhole camera: $\boldsymbol{\pi} : \mathbb{R}^3 \mapsto \Omega; \mathbf{X} \mapsto \mathbf{x}$ where $\Omega \subset \mathbb{R}^2$ is the domain of an image $I(\mathbf{x})$ and $\mathbf{x} = [x, y]^T = [X/Z,\ Y/Z]^T$ denotes coordinates in Ω. $S \subset \mathbb{R}^3$ is a smooth surface representing the shape of interest and $\mathbf{N} = [N_1, N_2, N_3]^T$ denotes the outward unit normal to S at each point $\mathbf{X} \in S$. Let $R = \boldsymbol{\pi}(S) \subset \Omega$ be the region on which the surface S is projected and $R^c = \Omega \setminus R$ be the complementary region of R. Similarly, the curve $\hat{c} = \boldsymbol{\pi}(C) \subset \Omega$ is the projection of the curve $C \subset S$ and $\hat{c}$ also denotes a boundary of R, $\hat{c} = \partial R$. Note, the curve $\hat{c}$ in 2D and the curve C in 3D are referred to as the "silhouette" and the "occluding curve", respectively.

4.3 Visual Pose Tracking with Monte Carlo Sampling on SE(3)

The basic idea of the method proposed in this section is to find the best transformation matrix so that the projection of its transformed model correctly coincides with 2D segmentation of the object of interest. To do this, we construct the first-order autoregressive model on a transformation matrix to predict the object's pose and a region-based energy model to optimally evaluate the projected model (or curve) in a 2D image plane.

The proposed algorithms in this section are related to the work in [25]. In [25], to find the optimal pose of an object of interest, the method of [25] obtains optimal pose parameters by gradient flow. Compared with [25], in this work, we randomly generate the set of 3D transformation matrices and then evaluate the proposed statistical energy model for each transformation matrix. To construct 3D transformation matrices, Monte Carlo sampling of the matrix group SE(3) is used, which is inspired by a filtering scheme on Lie groups introduced in [55]. This approach takes into account the underlying geometry of SE(3) rather than employing a local coordinate-based sampling because the set of three-dimensional transformations is not closely related to a vector space, but curved Lie groups [55]. The authors in [55] sequentially draw the set of two-dimensional affine group from a particle filtering and propagate it via an autoregressive process developed for state dynamics. Our work addresses a more general case of the work in [55] since we focus not only on tracking 2D motion but also on estimating a 3D pose.

4.3.1 Transformation Matrix

Transformation g is represented as in homogeneous coordinates:

$$g = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix}. \tag{41}$$

It is parameterized by the arbitrary parameter p as follows:

$$\begin{aligned} g_1 &= \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad g_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(p) & -\sin(p) & 0 \\ 0 & \sin(p) & cos(p) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ g_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & p \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad g_5 = \begin{bmatrix} \cos(p) & 0 & \sin(p) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(p) & 0 & \cos(p) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ g_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & p \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad g_6 = \begin{bmatrix} \cos(p) & -\sin(p) & 0 & 1 \\ \sin(p) & \cos(p) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{42}$$

After differentiating the above transformations in (42) with respect to parameter p at $p = 0$, $\frac{d\mathbf{g}_j}{dp}|_{p=0}$ $(j = 1, ..., 6)$, we obtain the Lie generators, B_j $(j = 1, ..., 6)$. Lie generators for 3D rigid motion span the tangent space to the Lie group manifold (see [6] for the detail derivation). Each generator corresponds to each geometric transformation, i.e., translations along x, y, and z axis,

rotations about the x, y, and z axes, respectively, given by:

$$B_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{43}$$
$$B_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_6 = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

4.3.2 Motion Model

In general, information we have on the dynamics of a system is incomplete. One way to overcome the difficulties associated with this lack of information is to use a stochastic model for the system dynamics. One can choose a random walk model for its simplicity. However, since the random walk model requires a sufficiently large number of samples to be practical; a Newtonian motion model [108] or an infinitesimal constant velocity model [55] can be chosen for a more efficient way to reach the optimal solution. In this work, by extending the work in [55], we construct the first-order autoregressive (AR) model on SE(3) based on motion variation to determine how a transformation matrix is generated, which is given by

$$g_k = g_{k-1} \exp\left(a \log(g_{k-2}^{-1} g_{k-1}) + du_k \sqrt{\Delta k}\right) \tag{44}$$

where $\exp(\cdot)$ and $\log(\cdot)$ denote the matrix exponential and the matrix logarithm, respectively [68]. a in (44) is the AR process parameter and Δk is a discrete time interval (Note that k denotes a discrete time step in only Section 4.3 for notational convenience). And, du_k is process noise determined by the degree of variation of an object motion and the Lie generators B_j ($j = 1, ..., 6$) as follows:

$$du_k = \sum_{j=1}^{6} \rho_{(j_{\mathbf{th}})} \tau_{(j_{\mathbf{th}})} B_j, \quad \tau \sim \mathcal{N}(0,\ \sigma_k) \tag{45}$$

where $\mathcal{N}(\cdot)$ represents the normal distribution and $\rho_{(j_{\mathbf{th}})}$ and $\tau_{(j_{\mathbf{th}})}$ denote the $j_{\mathbf{th}}$ element of $\rho \in \mathbb{R}^6$ and $\tau \in \mathbb{R}^6$, respectively. ρ is a user-defined diffusion weight vector; it controls the range of transformations in which we generate samples. In our experiments, $\rho = [0.5,\ 0.5,\ 0.5,\ 0.1,\ 0.1,\ 0.1]^T$ is used; but it can be relaxed by searching a larger space with the computational burden. To compute the covariance matrix σ_k, we represent g as a six-dimensional vector by using twist coordinates for

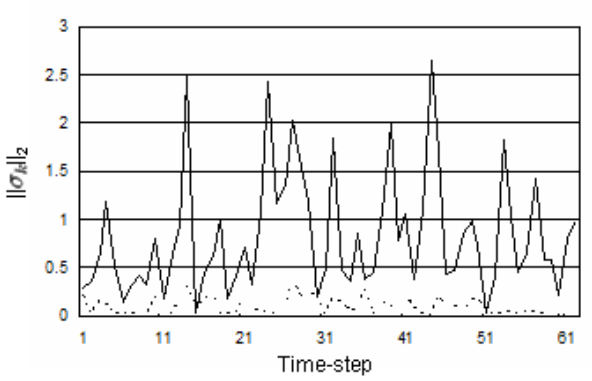


Figure 24: Graph of motion variation. A dotted line and a solid line denote the covariance variation of the sequences given in Figure 26(a) and Figure 26(b), respectively.

rotation $\mathbf{R}$ and attaching translation $\mathbf{T}$: $\lambda = [t_x, t_y, t_z, \omega_x, \omega_y, \omega_z]$. Thus, the covariance matrix $\sigma_k \in \mathbb{R}^{6\times6}$ is defined as:

$$\sigma_k = E\left[(\hat{\lambda}^i_{k-1} - \lambda_{k-1})(\hat{\lambda}^i_{k-2} - \lambda_{k-2})^T\right] \tag{46}$$

where $\hat{\lambda}^i (i = 1, ..., N)$ are parameters of the predicted transformation matrix in the Monte Carlo framework, which is described in Section 4.4.

The AR process in (44) determines the distribution of samples in the sample space according to motion variance of the tracked object during tracking. For example, the elephant in the sequence shown in Figure 26(a) is placed in the same position throughout the sequence. On the other hand, the elephant of the sequence shown in Figure 26(b) dynamically changes its pose due to a moving camera. The graph of Figure 24 shows that the degree of object's motion variation of the sequence in Figure 26(a) is much smaller than the sequence in Figure 26(b). Naturally, the degree of computed variance for a moving object is much higher than for a stationary object.

4.3.3 Energy Model

The proposed energy model to evaluate the optimality of a candidate transformation is simply defined as:

$$E_R = \int_R r_i(I(\mathbf{x}), \hat{c})\, d\Omega + \int_{R^c} r_o(I(\mathbf{x}), \hat{c})\, d\Omega \tag{47}$$

where $r_i : \chi,\ \Omega \mapsto R$ and $r_o : \chi,\ \Omega \mapsto R$ are mapping functions to measure the quality over inside (R) and outside (R^c) the curve, respectively. Here, χ is the space that corresponds to photometric

variable of interest. Intuitively, we want to drive the pose estimation so that the projected curve's interior and exterior distributions are maximally different. Any r_i and r_o functions that statistically describe the region properties of R and R^c can be used. For example, one can choose the mean intensity model or the distinct Gaussian model. In our work, r_i and r_o are defined as:

$$r_i = \log(\Sigma_i) + \frac{(I(\mathbf{x}) - \mu_i)^2}{\Sigma_i}, \quad r_o = \log(\Sigma_o) + \frac{(I(\mathbf{x}) - \mu_o)^2}{\Sigma_o}$$

where Σ_i and Σ_o are covariance, and μ_i and μ_o are intensity averages for R and R^c, respectively ($\Sigma_{i/o} \in \mathbb{R}^{3\times3}$ and $\mu_{i/o} \in \mathbb{R}^3$ for a color image).

Now, we assume that the statistical characteristic of the segmented object is not changing between consecutive frames. This assumption allows the tracker to find the curve with the most similar statistical property to the previously selected curve. Thus, we adopt the Bhattacharyya distance to evaluate the degree of similarity between intensity distributions of the tracked object in consecutive frames. The Bhattacharyya distance between two probability density functions, $p_1(z)$ and $p_2(z)$ with $z \in \mathbb{R}^N$, is defined by [49]

$$D_B = -\log\left(\int_{\mathbb{R}^N} \sqrt{p_1(z)p_2(z)}dz\right). \tag{48}$$

Now, we represent p_1 and p_2 as a normalized histogram h_1 and h_2, respectively. Histograms are relatively invariant to object motion, i.e., translation, rotation, and scale. Thus, metrics based on histograms can be good indicators of similarity between two distributions, for example, the intensity distributions of an object in consecutive video frames. A histogram can be computed by concatenating or by averaging histograms of each pre-defined color channel for color-based imagery. For example, one can obtain a histogram by using only pixels with large saturation and value in the HSV color space [82]. In our work, we use pixels inside $\hat{c}$ for histogram computation in which we average histograms of each RGB channel. Now, given two histograms, $h_1(b)$ and $h_2(b)$ with $b \in \mathbb{R}$, we have the following similarity distance measure using (48):

$$E_B = 1 - \int_{\mathbb{R}} \sqrt{h_1(b)h_2(b)}db. \tag{49}$$

The E_B is within [0, 1] and as the degree of similarity between histograms increases, as E_B goes to zero. Now, the optimal energy model is defined by adding two energy terms from (47) and (49):

$$E_{image} = E_R + E_B. \tag{50}$$

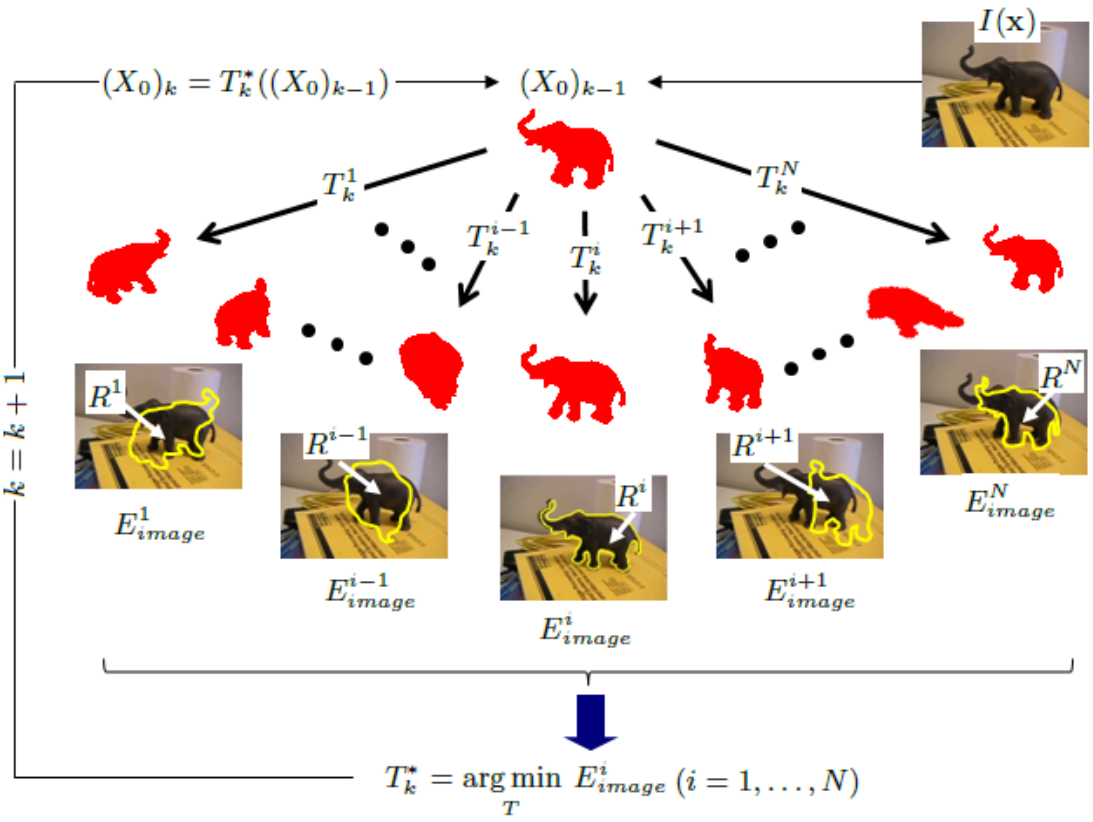


Figure 25: Schema summarizing the proposed 2D-3D pose tracking framework described in Section 4.4.

Therefore, the optimal projection (or projected curve) minimizes the E_{image}.

4.4 Tracking Framework I

We summarize the overall tracking approach by embedding the algorithms previously proposed in Section 4.3 into a Monte Carlo framework.

(a) Initialize $g_{k=0,1}$ and $h_{k=0,1}$.

(b) Generate N sample transformation matrices $\{g_k^i\}_{i=1}^N$ from (44):

$$g_k^i = g_{k-1} \exp\left(a \log(g_{k-2}^{-1} g_{k-1}) + du_k \sqrt{\Delta k}\right).$$

Note that transformation matrix at a current step k depends on transformations until two step behind, $k-1$ and $k-2$. Also, g_k^i refers to the estimated transformations of the previous time steps, g_{k-1} and g_{k-2}, not each $i_{\mathbf{th}}$ sample. This is different from the work of [55] based on particle filtering.

(c) Perform 3D transformation on X_0: $X^i = g_k^i(X_0)$.

(d) Project the transformed 3D model onto a 2D image plane: $R^i = \boldsymbol{\pi}(X^i)$ and compute the histogram of R^i, h^i.

(e) Evaluate the energy in (50) for each projected model: $E_{image}^i = E_R(R^i, I_k) + E_B(h_{k-1}, h^i)$.

(f) Estimate the optimal transformation using $g_k^* = \arg\min_g \ E_{image}^i$.

(g) Do iterations from step (b) to step (f) until E_{image} converges or the energy difference between frames is small enough.

(h) Let the histogram of the selected model be h_k and go to step (b): $k = k + 1$.

The schema of the proposed tracking framework is shown in Figure 25.

4.5 Experiments I

The proposed tracking algorithm was tested on several illustrative sequences in which color-valued images are used. The number of the generated transformation matrices is held fixed at $N = 100$ across all experiments. As the number of transformation matrices increases, more accurate results are expected, but the computational complexity increases exponentially. Figure 26 shows tracking results of the several sequences in noisy and cluttered environments. In particular, all sequences in Figure 26 capture the drastic changes of an objects' pose due to a moving camera except Figure 26(a). As one can see in Figure 26, successful pose tracking were obtained despite the dynamic pose change and the cluttered background. Note that severe scale changes of the red car were observed in Figure 26(d).

4.6 Particle Filters and Occlusion Handling for Rigid 2D-3D Pose Tracking

In this section, we extend the framework described in [25] by incorporating a particle filter for 2D-3D pose tracking. We also present an occlusion handling scheme to maintain the track of the target. In work of [25], the authors derive a variational approach to jointly carry out tasks of 2D region based segmentation and 3D pose estimation. This method shows robust performance for segmenting a 2D image and estimating the 3D pose of an object over image sequences even in cluttered

Figure 26: Tracking (a), (b) and (c) a grey elephant, and (d) a red car in noisy and cluttered environments. A camera is fixed in (a) and moving in (b), (c), and (d).

environments. However, since this method ignores the temporal nature of the observed images, it cannot handle erratic movements or challenging occlusions. That is, the variational technique relies only on image information to drive the corresponding 3D pose estimate, which may cause unsatisfactory results in the presence of occlusions that are statistically different from that of the object of interest; see the experiments in Section 4.8.2. Thus, in this section, we incorporate a particle filtering scheme in conjunction with the method in [25] and develop an occlusion handling scheme to exploit the underlying dynamics of the temporally observed data so that the proposed tracker continuously tracks the object of interest in a more general and challenging environment.

In addition, in this section, the variational approach of [25] is adopted in designing a measurement model to reduce computational complexity and to facilitate the effective search of a local optimum in a particle filtering framework. This method allows the samples to move further into modes

of a filtering distribution so that only a few samples are necessary; see Section 4.6.4. Variational methods, such as Mean-shift [17], are typically gradient based optimization methods minimizing a cost functional in order to find the local mode of a probability distribution. To effectively reduce the sample size of the particle filtering framework, variational approaches are embedded into particle filters in a number of works. The authors in [103] present a mean shift embedded particle filter, in which a smaller number of samples is used to estimate the posterior distribution than conventional particle filters by shifting samples to their neighboring modes of the observation so that samples are moved to have large weights. In [39], the underlying probability density function (pdf) is represented with a semi-parametric method using a mean-shift based mode seeking algorithm to solve a tracking problem for high dimensional color imagery. The authors of [109] fuse a deterministic search based on gradient descent and random search guided by a stochastic motion model, then, in object tracking, they effectively switch two search methods according to a momentum using the inter-frame motion difference.

The proposed algorithms are also related to the work in [60], which was described in previous sections from Section 4.3 to Section 4.5 in this chapter. In the work of [60], 3D transformation matrices are constructed by Monte Carlo sampling on a special Euclidean group in a motion model, and then a region-based statistical energy is applied to evaluate the optimality of each transformation matrix in a measurement model. Compared with the approach of [60], in this section, we employ a more natural particle filtering scheme to generate and propagate the translation and rotation parameters in a decoupled manner in order to find the optimal pose of an object of interest. In addition, we focus more on occlusion detection and handling in order to maintain track in the presence of significant occlusions by complex object motion. To this end, we improve the l-iteration scheme introduced in [88] and [95] by controlling dependencies between predictions and measurements of the system. This improvement provides the robust method to deal with occlusions of an obstacle with different statistical properties in a tracking framework based on region properties of an image.

4.6.1 Energy Functional and Gradient Flow

In this subsection, we briefly introduce the energy model and its gradient descent flow presented in [25]. These are important cornerstones in designing a measurement model for our tracking

framework. For our particular segmentation problem, we seek to find a boundary that optimally partitions the object of interest or foreground from the corresponding background in a given 2D image. Inspired by region-based active contour models [13, 81, 73], the authors in [25] define an objective energy functional based on the global statistics of an image so that the curve $\hat{c}$ (and 3D pose) is evolved to maximize the image statistical measure of discrepancy between its interior and exterior regions. This is given as follows:

$$E = \int_R r_o(I(\mathbf{x}), \hat{c}) d\Omega + \int_{R^c} r_b(I(\mathbf{x}), \hat{c}) d\Omega \tag{51}$$

where $r_o : \chi, \Omega \mapsto \mathbb{R}$ and $r_b : \chi, \Omega \mapsto \mathbb{R}$ are functions measuring the visual consistency of the image pixels with a statistical model over the regions R and R^c, respectively. Here, χ is the space that corresponds to photometric variable of interest. In this work, r_o and r_b are given by:

$$r_o = -\log(\Sigma_o) - \frac{(I(\mathbf{x}) - \mu_o)^2}{\Sigma_o}, \quad r_b = -\log(\Sigma_b) - \frac{(I(\mathbf{x}) - \mu_b)^2}{\Sigma_b}. \tag{52}$$

Here Σ_o and Σ_b are variances inside and outside the curve $\hat{c}$, and are given by

$$\Sigma_{\mathrm{o}} = \frac{\int_R \big(I(\mathbf{x}) - \mu_{\mathrm{o}}\big)^2 d\Omega}{\int_R d\Omega}, \quad \Sigma_{\mathrm{b}} = \frac{\int_{R^c} \big(I(\mathbf{x}) - \mu_{\mathrm{b}}\big)^2 d\Omega}{\int_{R^c} d\Omega} \tag{53}$$

where

$$\mu_{\mathrm{o}} = \frac{\int_R I(\mathbf{x}) d\Omega}{\int_R d\Omega}, \quad \mu_{\mathrm{o}} = \frac{\int_{R^c} I(\mathbf{x}) d\Omega}{\int_{R^c} d\Omega}. \tag{54}$$

For gray-scale images, $\mu_{\mathrm{o/b}}$ and $\Sigma_{\mathrm{o/b}}$ are scalars and for color images, $\mu_{\mathrm{o/b}} \in \mathbb{R}^3$ and $\Sigma_{\mathrm{o/b}} \in \mathbb{R}^{3\times 3}$ are vectors and matrices. Note that r_o and r_b can be chosen as various forms describing the region properties of the pixels located inside and outside the curve (e.g., mean intensities [13], distinct Gaussian densities [81], and generalized histograms [73]). As seen above, r_o and r_b are chosen to be the region based functional of [81].

Now, let $X_0 \in \mathbb{R}^3$ be the coordinates of points on S_0 where S_0 is the identical reference surface in 3D. By the rigid transformation $g \in SE(3)$, one can locate S in the camera referential by $S = g(S_0)$. Written in a point wise fashion yields $\mathbf{X} = g(\mathbf{X}_0) = \mathbf{R}\mathbf{X}_0 + \mathbf{T}$ with $\mathbf{R} \in SO(3)$ denoting the rotational group and $\mathbf{T} \in \mathbb{R}^3$ representing translations. Here, 3D pose motions are represented by a set of six parameters. The parameters of the rigid motion g will be denoted by $\lambda = [\lambda_1, \ldots, \lambda_6]^T = [t_x, t_y, t_z, \omega_x, \omega_y, \omega_z]^T$. Rotations are represented in exponential coordinates,

which is a more compact form than using quaternion (4 entries) or basic rotation matrices in three dimensions (12 entries); see [68]. Now, since we assume that the 3D shape of the rigid object is known, our objective is to minimize energy E in (51) by exploring only the regions R and R^c that result from projecting the surface S onto the image plane. For a calibrated camera, these regions are functions of the transformation g only. Solving for the transformation that minimizes E can be undertaken via gradient descent over the parameters λ. This is described next.

The partial differentials of E with respect to the pose parameters λ_i's can be computed using the chain-rule:

$$\frac{\partial E}{\partial \lambda_i} = \int_{\hat{c}} \Big(r_o(I(\mathbf{x})) - r_b(I(\mathbf{x})) \Big) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} + \int_R \left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega + \int_{R^c} \left\langle \frac{\partial r_b}{\partial \hat{c}} \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \tag{55}$$

where $\hat{s}$ is the arc-length parameterization of the silhouette $\hat{c}$ and $\hat{\mathbf{n}}$ is the (outward) normal to the curve at $\mathbf{x}$.

Using the arc-length s of C and $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, one has

$$\begin{aligned} \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} &= \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, J \frac{\partial \hat{c}}{\partial \hat{s}} \right\rangle d\hat{s} = \left\langle \frac{\partial \boldsymbol{\pi}(C)}{\partial \lambda_i}, J \frac{\partial \boldsymbol{\pi}(C)}{\partial s} \right\rangle ds \\ &= \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix} \frac{\partial \mathbf{X}}{\partial s} \right\rangle ds \\ &= \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds = \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_{\mathbf{X}} \kappa_{\mathbf{t}}}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i} \mathbf{N} \right\rangle ds \end{aligned} \tag{56}$$

where K denotes the Gaussian curvature, and $\kappa_{\mathbf{X}}$ and $\kappa_{\mathbf{t}}$ denote the normal curvatures in the directions $\mathbf{X}$ and $\mathbf{t}$, respectively, where $\mathbf{t}$ is the vector tangent to the curve C at the point $\mathbf{X}$, i.e. $\mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}$. Note that the last two terms in (55) collapse due to the choice of the r_o and r_b in (52). Now we have the following gradient descent flow (see Appendix C and [25] for details):

$$\frac{\partial E}{\partial \lambda_i} = \int_C \Big(r_o\big(I(\boldsymbol{\pi}(\mathbf{X}))\big) - r_b\big(I(\boldsymbol{\pi}(\mathbf{X}))\big) \Big) \cdot \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_{\mathbf{X}} \kappa_{\mathbf{t}}}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds \tag{57}$$

where the term $\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle$ can be computed for the evolution of the pose parameter λ_i which is a translation parameter ($i = 1, 2, 3$) or a rotation parameter ($i = 4, 5, 6$):

- For a translation parameter,

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X}_0 + \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle$$
$$= \left\langle \begin{bmatrix} \frac{\partial \lambda_1}{\partial \lambda_i} \\ \frac{\partial \lambda_2}{\partial \lambda_i} \\ \frac{\partial \lambda_3}{\partial \lambda_i} \end{bmatrix}, \mathbf{N} \right\rangle = \left\langle \begin{bmatrix} \delta_{1,i} \\ \delta_{2,i} \\ \delta_{3,i} \end{bmatrix}, \mathbf{N} \right\rangle = N_i \tag{58}$$

  where the Kronecker symbol $\delta_{i,j}$ was used ($\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ otherwise) and $\mathbf{T} = [t_x, t_y, t_z]^T = [\lambda_1, \lambda_2, \lambda_3]^T$.

- For a rotation parameter,

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X_0}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \mathbf{R} \begin{bmatrix} 0 & -\delta_{6,i} & \delta_{5,j} \\ \delta_{6,i} & 0 & -\delta_{4,i} \\ -\delta_{5,i} & \delta_{4,i} & 0 \end{bmatrix} \mathbf{X}_0, \mathbf{N} \right\rangle \tag{59}$$

  where $\mathbf{R} = \exp\left(\begin{bmatrix} 0 & -\lambda_6 & \lambda_5 \\ \lambda_6 & 0 & -\lambda_4 \\ -\lambda_5 & \lambda_4 & 0 \end{bmatrix} \right)$.

4.6.2 State Space Model

In what follows, the location of the object can be characterized by the translation and rotation parameters of a rigid transformation. Thus, the (hidden) variable or state s_t that we want to estimate are the pose parameters at time t and is given by

$$s_t = \begin{pmatrix} \mathbf{T} \\ \mathbf{W} \end{pmatrix}_t . \tag{60}$$

Here, the translation and rotation vectors are $\mathbf{T} = [t_x, t_y, t_z]^T$ and $\mathbf{W} = [\omega_x, \omega_y, \omega_z]^T$, respectively. We should mention that many 2D visual tracking schemes involving elastic deformations of the target have a theoretically infinite dimensional state space [24, 87]. In contrast, the state variable describing the (perceived) deformation in the 2D domain can now be succinctly represented via a finite set in the 3D space. Note that this only holds for the particular but general case of tracking rigid objects. Lastly, when a new image I_t arrives at time t, we obtain an observation z_t.

4.6.3 Prediction Model

As mentioned previously, it is important to carefully utilize system dynamics in order for a given tracking algorithm to converge to the correct optimum. One may choose a random walk model for the state transition equation. However, since this model is usually only practical with a sufficiently large number of samples in a particle filtering framework, we chose instead to employ a first-order autoregressive (AR) model [9] for the state dynamics. We perform the propagation of translation and rotation parameters in a decoupled manner. Consequently, the system dynamics for predicting pose parameters, $\hat{s}_t = [\hat{\mathbf{T}}_t, \hat{\mathbf{W}}_t]^T$, is given by:

$$\begin{aligned} \hat{\mathbf{T}}_t^i &= \mathbf{T}_{t-1} + A(\hat{\mathbf{T}}_{t-1}^i - \mathbf{T}_{t-1}) + u_t^i \\ \hat{\mathbf{W}}_t^i &= \mathbf{W}_{t-1}^i + u_t^i. \end{aligned} \tag{61}$$

The rotation matrix R for each particle $\{s_{t-1}^i\}_{i=1,\ldots,N}$ can be computed by $\hat{R}_t^i = \exp(\hat{W}_t^i) \cdot R_{t-1}$ where $\exp(\cdot)$ denotes the matrix exponential [68] and A is the AR process parameter. The noise model u_t^i is defined as:

$$u_t^i \sim \mathcal{N}(0,\ \rho \cdot e_{t-1}^i (e_{t-1}^i)^T) \tag{62}$$

where $\mathcal{N}(\cdot)$ represents the normal distribution and ρ is a user-defined diffusion weight. Moreover, a motion alignment error e_{t-1}^i for each particle $\{s_{t-1}^i\}_{i=1,\ldots,N}$ is obtained from the predicted and measured states at time $t-1$, i.e.,

$$e_{t-1}^i = \tilde{s}_{t-1}^i - \hat{s}_{t-1}^i \tag{63}$$

where $\tilde{s}_{t-1} = [\tilde{\mathbf{T}}_{t-1}, \tilde{\mathbf{W}}_{t-1}]$ is the measured state vector at time $t-1$. Here, it is noted that the proposed decoupled methodology not only provides flexibility in designing the state dynamics of the system, but it also allows us to differently deal with the translational and rotational equations in (61) by controlling the diffusion weight. For example, since an orientation space is wrapped on itself and angles behave linearly within small angle approximation [8], we apply a small perturbation to only rotational dynamics without affecting the translation equation because of the decoupling.

Now, inspired by [95], we define the bandwidth b_{t-1}^i for each particle as the combination of the diffusion weight ρ and the motion alignment error e_{t-1}^i. That is, $b_{t-1}^i = \ \rho \cdot e_{t-1}^i (e_{t-1}^i)^T$. Then

the process noise can be represented by a multivariate Gaussian distribution based on the bandwidth term, b_{t-1}^i:

$$u_t \sim \frac{1}{N}\sum_{i=1}^{N} \frac{1}{(2\pi)^{N/2}|b_{t-1}^i|^{1/2}} e^{-\frac{1}{2}(s_t^i - s_{t-1}^i)^T (b_{t-1}^i)^{-1}(s_t^i - s_{t-1}^i)}. \tag{64}$$

From (64), one can see that the particles are driven by the bandwidth term in an online fashion and diffuse in the direction of motion of the object. Next, we discuss the measurement model.

4.6.4 Measurement Model

Now we specify the measurement function $h(\cdot)$ for the observed image I_t at time t as follows. First, we carry out l-iterations of gradient descent flow in (57) for each particle $\{\hat{s}_t^i\}_{i=1,\ldots,N}$. Here, it should be noted that the choice of l is carefully considered to avoid *sample degeneracy* and *sample impoverishment*. If the tracker reaches a local minimum of the objective functional in (51) with too large l, the state at time t would lose the temporal coherency with the state at time $t-1$ (*sample degeneracy*). On the other hand, if l is selected to be too small, most particles would not move to the region of high likelihood (*sample impoverishment*). While the authors in [88] and [95] use a l-iteration scheme for their resampling algorithm, l is experimentally chosen based on the imagery and the type of local optimizer used. In the present work, contrary to previous works, the number of l is dynamically chosen online according to the degree of occlusion and object's movement. This will be described in Section 4.6.5.

Next, we assign and update the importance weights associated for each particle. This is done by employing a transitional prior density as the proposal distribution [90]. In doing so, the weight update equation is $w_t^i = w_{t-1}^i p(z_t \mid s_t)$ where the likelihood of the observation is defined as:

$$p(z_t \mid s_t) = e^{-E(\mathbf{T}_t,\, \mathbf{W}_t,\, I_t)}. \tag{65}$$

Lastly, we obtain the measurement for time t. That is, the pose parameters with the minimum energy are taken as the measurement pose. Thus, the projected silhouette is the best fitting curve which makes its interior and exterior statistical properties be maximally different in a 2D image domain (i.e., it minimizes the region-based energy in (51)).

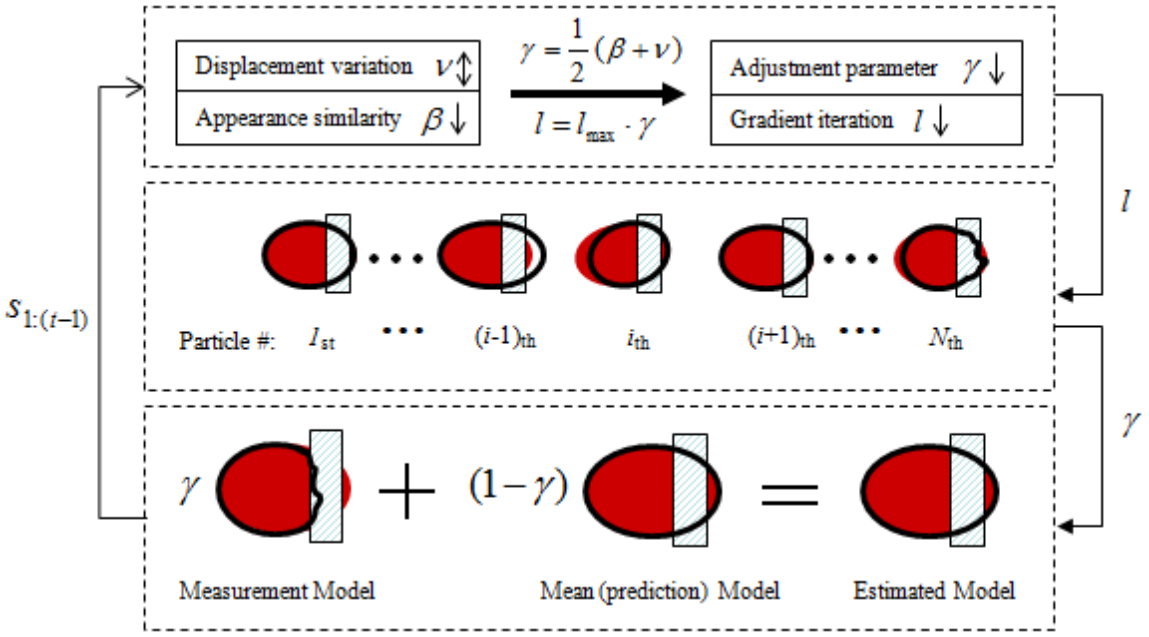


Figure 27: Schema summarizing the proposed occlusion handling scheme.

4.6.5 Occlusion Handling

The presence of occlusions generally hinders tracking algorithms from continuously tracking an object of interest. Occlusion detection is a necessary task before performing occlusion handling. In [124], the authors detect an occlusion using the relative change of the object's size compared to the average object's size as well as the distance between the object and an obstacle. However, this method may give ambiguous results if an object's size changes due to camera zooming. Here, we propose a histogram based occlusion detection technique, which is performed by checking the variation of the histogram of the object during tracking. To do this, we define an appearance model as a normalized histogram h. For color-based imagery, a histogram is calculated by the mean color (in an RGB color space) of the pixels inside $\hat{c}$. Here, the RGB color space is normalized to remove the effect of intensity variations. Note that better performance could be expected if one uses other color spaces or image feature descriptors such as HSV, $m_1m_2m_3$ spaces [35], spatiograms [7], and HOG (histograms of oriented gradients) [22]; see [35, 7, 22] and references therein. However, we chose the normalized histogram in the normalized RGB space to construct the appearance model due to their simplicity and usability.

The evaluation of the histogram change is achieved by computing the Bhattacharyya coefficient

between two appearance models of the current silhouette curve and of the template model. The template model can be obtained from the initial curve of the first segmentation of the given sequence. The Bhattacharyya distance [49] between two probability density functions (pdfs), p_1 and p_2, is defined by:

$$D_B = -log(\int_{\mathbb{R}^2} \sqrt{p_1(\mathbf{x})p_2(\mathbf{x})}d\mathbf{x}). \tag{66}$$

Considering discrete densities, such as $h_t(b)$ and $h_{template}(b)$, the Bhattacharyya coefficient is defined as:

$$\beta = \sum_{b\in\mathbb{R}} \sqrt{h_t(b)h_{template}(b)} \tag{67}$$

where $h_t(b)$ and $h_{template}(b)$, with $b \in \mathbb{R}$, are appearance models of the curve $\hat{c}$ at time t and of a template model, respectively. The Bhattacharyya coefficient β varies between 0 and 1 (0 indicates complete mismatch and 1 is a perfect match). Small β indicates that another object has occluded the target being tracked because statistical information inside the tracked object is changed. Thus, in this work, the Bhattacharyya coefficient β is used as the appearance similarity measure between two histograms. Unfortunately, the template model can be influenced by illumination changes and geometric variations as well as through camera angle differences even though no occlusion occurs. To cover the undesired appearance changes, we should periodically update the template model according to the degree of histogram variations between frames. More specifically, if the histogram of the target is changing slowly over time, the template model is updated as the new histogram of the segmented object at a current frame and is preserved otherwise. The update condition for the template model is defined as

$$\beta(h_{template},\ h_{(t-1)t_d}) > \beta_{th}, \quad (t > 1) \tag{68}$$

where β_{th} is a positive threshold between 0 and 1. The value t_d is the user-defined checking interval. Note that the interval for checking should be set large enough because the histogram variance between consecutive frames is generally small even though the occlusion occurs. This approach allows the tracker to keep the histogram of the template model until the sequence finishes. In this work, $\beta_{\mathbf{th}}$ and t_d are experimentally selected as $[0.7, 0.8]$ and $[10, 20]$, respectively.

Now, we elucidate the proposed occlusion handling scheme. The basic idea of occlusion handling for our algorithm is regularization between the measurement and prediction model. To do

this, l-iterations of gradient descent of the objective functional in (57) is dynamically adjusted in an online manner. Not only is this l-iteration scheme used to resample the particles in Section 4.6.4, but also it can be interpreted as a function of the *Kalman gain* in a Kalman filtering like framework. In other words, one can view this parameter as the amount of confidence in the system model with respect to the current measurements [86]. For example, if the object being tracked is not observed appropriately during occlusion, then the degree of trust of the measurement model is reduced. Thus, we should only employ a few iterations within the measurement model so that the method depends more on the prediction model to maintain the track. On the other hand, if one can completely trust the obtained measurement (e.g., the observed image shows a smooth object movement without occlusions), the number of l should be assigned to be relatively large.

While the method proposed is similar to that of [86], a key difference is that we dynamically choose the number of l-iterations online based on both the degree of occlusion (or severity) and the degree of the object's motion displacement as opposed to an experimental fixed choice of l. In addition, the degree of object's pose variation is obtained from using the accumulated history of the object's location; translation and rotation vectors

$$\nu = 1 - \exp\{-\mathbf{var}(s_{(t-t_d):t})\} \tag{69}$$

where $\mathbf{var}(\cdot)$ is variance of the given variable and the checking interval, t_d, indicates how many previous states are used. If the variation of the location over the previous frames is large, we infer that the object is moving during the sequence and l is maintained so that the tracker is able to follow the object's movement as much as possible. Now the adjustment parameter γ is computed and the number of l-iterations is assigned online as follows:

$$\gamma = \frac{1}{2}(\beta + \nu), \qquad l = l_{max} \cdot \gamma \tag{70}$$

where l_{max} is a maximum allowance iteration number of l. This can also be viewed as a boundary and initial condition of l.

The overview of the proposed occlusion handling scheme is illustrated in Figure 27. Here, if the occlusion is detected, the adjustment parameter γ decreases according to the displacement variation ν and appearance similarity parameter β. Consequently, it reduces the number of l-iterations of the gradient descent flow. This approach eventually overcomes occlusions in the course of tracking

due to the fact that it makes the algorithm depend more on the prediction model in challenging situations.

4.7 Tracking Framework II

An overview of the proposed system for 2D-3D object tracking using the particle filtering framework proposed in Section 4.6 is now described.

1) Initialization Step:

   (a) Initialize state, s_t, at $t = 0$ by using 2D segmentation and 3D pose estimation method introduced in [25] in the first frame of the given sequence.

   (b) Obtain the template appearance model, $h_{template}(\mathbf{x})$, at $t = 0$ from the initial segmented curve in a 2D image plane.

2) Prediction Step:

   (a) Generate N transformation parameters, $\{\hat{s}_t^i\}_{i=1,\ldots,N}$, around s_{t-1}^i by (61):

$$\hat{\mathbf{T}}_t^i = \mathbf{T}_{t-1} + A(\hat{\mathbf{T}}_{t-1}^i - \mathbf{T}_{t-1}) + u_t^i$$
$$\hat{\mathbf{W}}_t^i = \mathbf{W}_{t-1}^i + u_t^i.$$

3) Update Step:

   (a) Perform l-iterations of the gradient descent flow in (57) on each generated parameter, $\{\hat{s}_t^i\}_{i=1,\ldots,N}$.

   (b) Calculate the importance weights from (65):

$$\tilde{w}_t^i = \tilde{w}_{t-1}^i e^{-E(\hat{\mathbf{T}}_t^i,\, \hat{\mathbf{W}}_t^i,\, I_t)}$$

   and normalize:

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{i=1}^N \tilde{w}_t^i}.$$

   (c) Represent the posterior distribution of the system by a set of weighted particles:

$$p(s_t \mid z_{1:t}) = \sum_{i=1}^{N} w_t^i \delta(s_t - s_t^i).$$

(d) Resample N particles according to $p(s_t \mid z_{1:t})$ by using the generic re-sampling scheme introduced in [90]. Note that $\{\overline{s}_t^i\}_{i=1\ldots N}$ denote resampled particles.

4) Adjustment Step:

(a) Estimate state, s_t, using the mean state of the set $\overline{s}_t^i$ and the measurement state s_t^m as follows:

$$s_t = \gamma s_t^m + (1-\gamma)(\frac{1}{N}\sum_{i=1}^{N}\overline{s}_t^i). \tag{71}$$

(b) Update the template appearance model, $h_{template}$, by the condition in (68) and compute the appearance similarity measure, $\beta(h_t,\ h_{template})$.

(c) Compute the adjustment parameter γ and assign the number of l by (70).

4.8 Experiments II

Various synthetic and real sequences of different rigid objects were used to demonstrate the robustness of the proposed method to noise, cluttered environments, as well as the algorithm's ability to cope with partial occlusions and imperfect information. In this section, we provide qualitative and quantitative results of various tracking scenarios including a comparison to the algorithms presented in [25] and [60]. In particular, in the quantitative experiments, two quantitative results regarding the robustness of noise and occlusion of the proposed method are provided on synthetic data. We also should note that because code of other joint 2D-3D pose estimation/segmentation algorithms were not readily available, our experiments are focused on highlighting the advantages and limitations of exploiting dynamics in visual tracking. However, before doing so, we briefly mention some numerical details associated with the experiments performed.

Implementation Details: In these experiments, the parameters used were held fixed across all sequences. Specifically, the value of maximum l-iteration, l_{max}, was selected within $[20, 30]$. Its range is chosen by depending on the objective functional used and its step-size (e.g. here, the gradient descent flow in (57)). The number of particles $N = 40$ was empirically chosen to provide good performance without significant computational burden. Note that the setting of minimal amount of samples ($N = 40$) can be realized by embedding the gradient descent flow into measurement function proposed in Section 4.6.4. Using this scheme allows the particle filtering framework to rely on a small number of particles as opposed to the conventional CONDENSATION filter [44]. Note, this

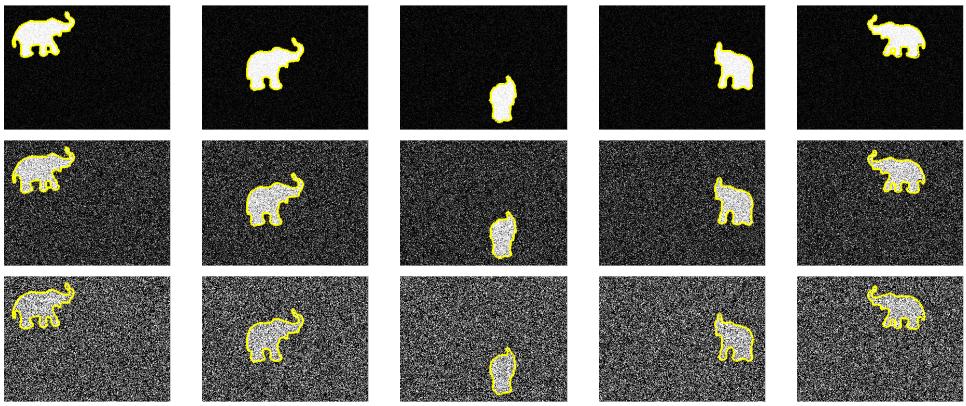


Figure 28: Quantitative tracking results for robustness test to noise over 200 frames of the sequences. Gaussian noises with $\sigma_n^2 = 1\%$ (upper row), $\sigma_n^2 = 25\%$ (middle row), and $\sigma_n^2 = 100\%$ (bottom row) were added, respectively.

Table 3: Table displaying $\%-$*absolute* error statistics for Gaussian noises with $\sigma_n^2 = 1\%$, $\sigma_n^2 = 25\%$, $\sigma_n^2 = 50\%$, $\sigma_n^2 = 75\%$, and $\sigma_n^2 = 100\%$ over 200 frames of the sequences as given in Figure 28. T.avg, T.std, R.avg, and R.std denote average and standard deviation of translation error and rotation error, respectively.

σ_n^2	$\%-$*absolute* error (in $\%$)			
	T.avg	T.std	R.avg	R.std
1%	3.01	0.74	3.56	3.29
25%	3.00	0.74	3.68	3.41
50%	3.05	0.86	3.86	3.48
75%	3.25	1.58	5.17	4.42
100%	3.52	1.68	5.09	4.65

setup is also used for exploration of shape deformation [88] and point set registration [95]. Overall, using un-optimized code for the proposed method shows acceptable performance with computation time of approximately 10 seconds per frame on a 3.6GHz Windows machine with 2GB of RAM.

4.8.1 Tracking in Noisy and Cluttered Environments

In this subsection, first of all, we show quantitative results regarding the robustness of the proposed method to noise on synthetic data. In generating the synthetic data, we first construct a basic elephant sequence, and then add several diverse noise levels of Gaussian noise whose variance ranges from $\sigma_n^2 = 1\%$ to $\sigma_n^2 = 100\%$. The sequences (and results) can be seen in Figure 28. The translation

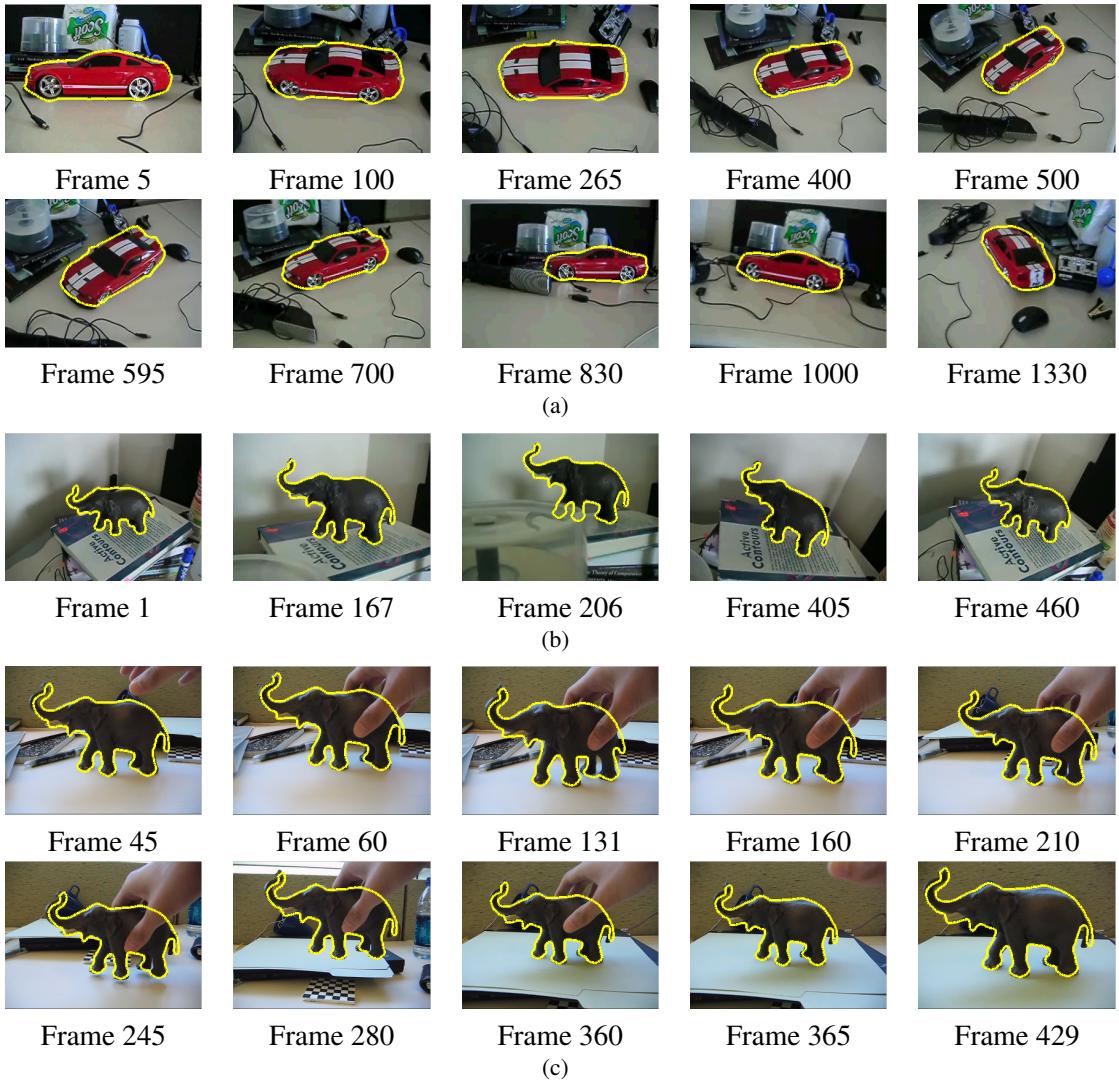


Figure 29: Tracking in noisy and cluttered environments.

and rotation parameters linearly increase and decrease throughout the sequences of 200 frames to produce a large variation for the aspect of the object. The size of the sequence images is 242×322. To quantitatively evaluate the tracking results, percent$(\%)$-*absolute* errors are computed for both translation and rotation over each level of noise sequences:

$$\% - \textit{absolute}\ \textbf{error} = \frac{\|\mathbf{v}_{\text{measured}} - \mathbf{v}_{\text{truth}}\|}{\|\mathbf{v}_{\text{truth}}\|} \times 100 \tag{72}$$

where $\mathbf{v}_{\text{measured}}$ and $\mathbf{v}_{\text{truth}}$ are measured and ground-truth of translation and rotation vectors, respectively. Table 3 displays average and standard deviation errors of position and rotation of the

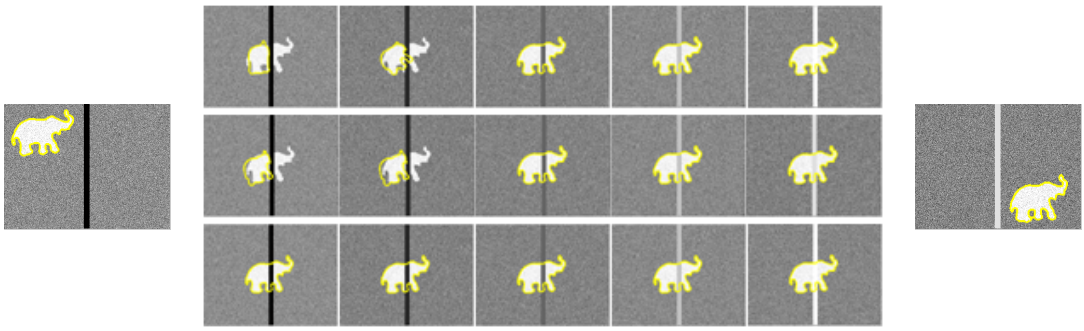
results from the sequences in Figure 28. As the noise level increases, the average and standard deviation errors also increase. This is due to the probability of encountering unexpected local minima will also increase given that it is now harder to distinguish the object of interest from the background. However, tracking is still maintained throughout the entirety of each sequence and the tracking errors did not exceed above 3% and 5% for translation and rotation, respectively. Note that the elephant and backgrounds are barely visibly distinguishable in the sequence of noise level $\sigma_n^2 = 100\%$.

Figure 29 shows several sequences capturing the drastic changes of an object's pose in a cluttered background. Despite the cluttered background and significant changes of pose, successful tracking results were obtained. The sequence in Figure 29(a) is comprised of 1350 frames, while Figure 29(b) contains 470 frames. For the sequence shown in Figure 29(c), there are 450 frames available to track. In Figure 29(a) and 29(b), the red car and the gray elephant are observed under a dynamically moving camera. The elephant is manually moved by a hand as one can see in Figure 29(c). Thus, the corresponding pose of all the objects tested is altered significantly over the various sequences. Nevertheless, the proposed algorithm yields satisfactory tracking results. In particular, the hand holding the elephant partially occludes the elephant in Figure 29(c). However, in this case, since the hand has the similar color to the elephant, it does not have much effect on the statistical property of the elephant, and the track of the elephant is successfully maintained. In the next subsection, we demonstrate the ability of the proposed occlusion handling scheme to deal with partial occlusions and imperfect information in the scenarios including the occlusions that contain a statistically different intensity from that of the object of interest.

4.8.2 Tracking in the Presence of Occlusion

In contrast to the previous sequences, the scenarios in this subsection not only include dynamic changes of an object's pose, but other objects (such as markers, staples, and a dark elephant), which occlude the object of interest, and provide an added difficulty to the overall tracking problem. Moreover, this subsection demonstrates how the proposed method outperformed the approaches of [25] and [60] in dealing with occlusion handling.

First, to provide quantitative results regarding the robustness of the proposed method in handling



Frame 2 Frame 65 Frame 130

Figure 30: Quantitative tracking results for robustness test to different brightness levels of the occlusion bar over 130 frames of the sequences. Gaussian noise with $\sigma_n^2 = 1\%$ was added. From left to right in frame 65, the brightness levels of the bar were assigned as 0.1, 0.3, 0.5, 0.7, and 0.9, respectively. Upper row in frame 65: results using the method in [25]. Middle row in frame 65: results using the method in [60]. Bottom row in frame 65: results using the proposed method. In frame 2 and frame 130, results using the proposed method are only displayed when the brightness levels of the bar were assigned as 0.1 and 0.9, respectively.

Table 4: Brightness Level Table: table displaying $\%-$*absolute* error statistics over 130 frames of the sequences as given in Figure 30. The indicators, *, $^\diamond$ and $^\#$, denote the results using the proposed method, using the method in [60], and using the method in [25], respectively. $\mathrm{T}^{(\cdot)}$ and $\mathrm{R}^{(\cdot)}$ denote the average values of translation error and rotation error, respectively. Note that no $\%-$*absolute* errors are obtained in case of the loss of track.

Brightness Level	$\%-$*absolute* error (in $\%$)					
	T^*	$\mathrm{T}^\diamond$	$\mathrm{T}^\#$	R^*	$\mathrm{R}^\diamond$	$\mathrm{R}^\#$
0.1	2.2	n/a	n/a	3.77	n/a	n/a
0.3	2.2	n/a	n/a	3.77	n/a	n/a
0.5	2.4	1.82	2.47	3.8	4.9	4.4
0.7	1.7	2.14	2.55	3.2	4.11	4.19
0.9	2.1	2.44	2.8	2.9	4.41	4.5

occlusions, we generate set of synthetic sequences, in which an obstacle bar exhibiting different levels of gray-scale intensity is added. In these sequences, Gaussian noise with zero mean and variance of $\sigma_n^2 = 1\%$ was added to a binary image of the toy elephant. From a tracking viewpoint, the bar prevents many algorithms, which rely only on image and shape information, from successfully maintaining track of the object. Specifically, we vary the brightness level of the bar from 0.1 to 0.9 where 0 denotes black and 1 denotes white. The results are shown in Figure 30. In Figure 30, when the brightness level of the bar is less than 0.5, the methods in [25] and [60] lose the track, but the

proposed tracking algorithm maintains track over the generated sequences. The average values of translation and rotation errors are measured by equation (72) and are displayed in Table 4. The table shows the improvement of the proposed method for occlusion handling compared to the methods in [25] and [60]. More detailed comparison of the proposed algorithms with the methods of [25] and [60] is discussed in the next experiments on real sequences.

Figures 31, 32, and 33 show a white marker (vertically or horizontally held by a hand) and a red marker (vertically held by a hand) that pass by the gray elephant from right to left in a cluttered background, respectively. We should mention that the algorithms in [25] works well with the occlusions that are similar in nature to that of the interested object; see Figure 30. However, due to the fact that the occlusion contains a statistically different intensity from that of the object of interest, the methods in [25] is not able to maintain track as shown in Figures 31(a), 32(a), and 33(a). Specifically, utilizing only the gradient descent flow presented in equation (57), the movement of the marker acts as if it "pushes" or "blows" the silhouette curve off of the elephant. This is simply due to the statistical difference between the object and the occlusion. That is, if one were to look at equation (57) from a robust statistics point of view, the flow regarding the integration about the occluding curve excludes the possible points on the red or white markers because they are viewed as outliers. In turn, one cannot properly estimate the 3D pose or maintain track. However, if one exploits the underlying dynamics as done in the proposed algorithm, one achieves a more robust result. In Figure 34, the helicopter is manually moved throughout several set of staples that are positioned at different heights. While the tracker in [25] fails after the helicopter passes by the second set of staples as shown in Figure 34(a), the proposed tracker overcomes the continual occlusions and maintains track of the pose of the helicopter.

It is interesting to note that the tracker of [60] eventually lost the track of the helicopter and got trapped in a local minimum as shown in Figure 34(b) even though it continuously estimated the pose of the elephant in the presence of occlusions (i.e., the red or white markers) in Figures 31(b), 32(b), and 33(b). These results show that the approach of [60] is more vulnerable to occlusions when it tracks a moving object than a stationary object. This is because the work of [60] disregards control of predictions and measurements of the system, which was taken into account in the present occlusion handling scheme. In addition, in the proposed method, in contrast to the work of [60], the

Figure 31: Elephant sequence I with occlusion in a cluttered environment. Tracking results (a) using the method in [25] (b) using the method in [60], and (c) using the proposed method.

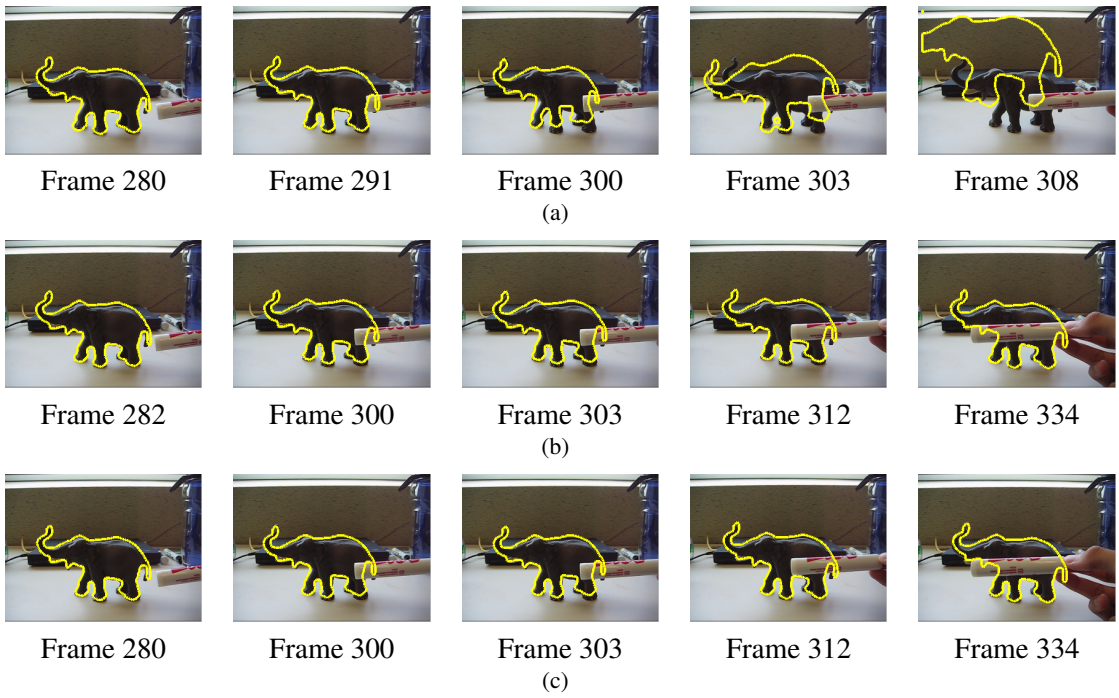


Figure 32: Elephant sequence II with occlusion in a cluttered environment. Tracking results (a) using the method in [25], (b) using the method in [60], and (c) using the proposed method.

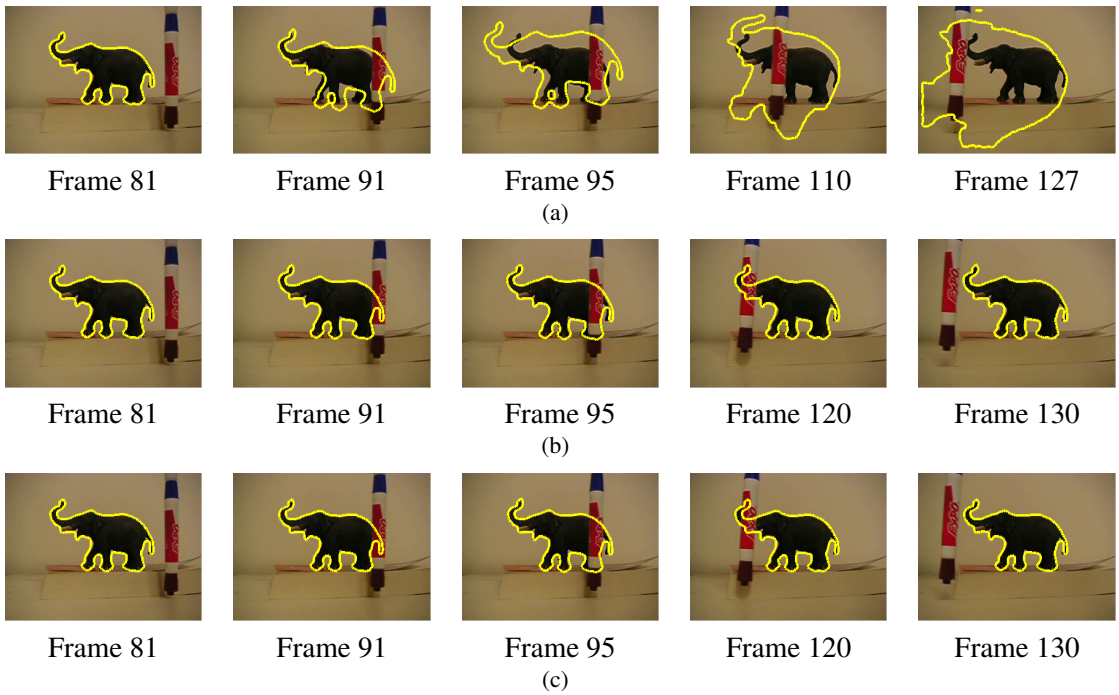


Figure 33: Elephant sequence I with occlusion in a cluttered environment. Tracking results (a) using the method in [25] (b) using the method in [60], and (c) using the proposed method.

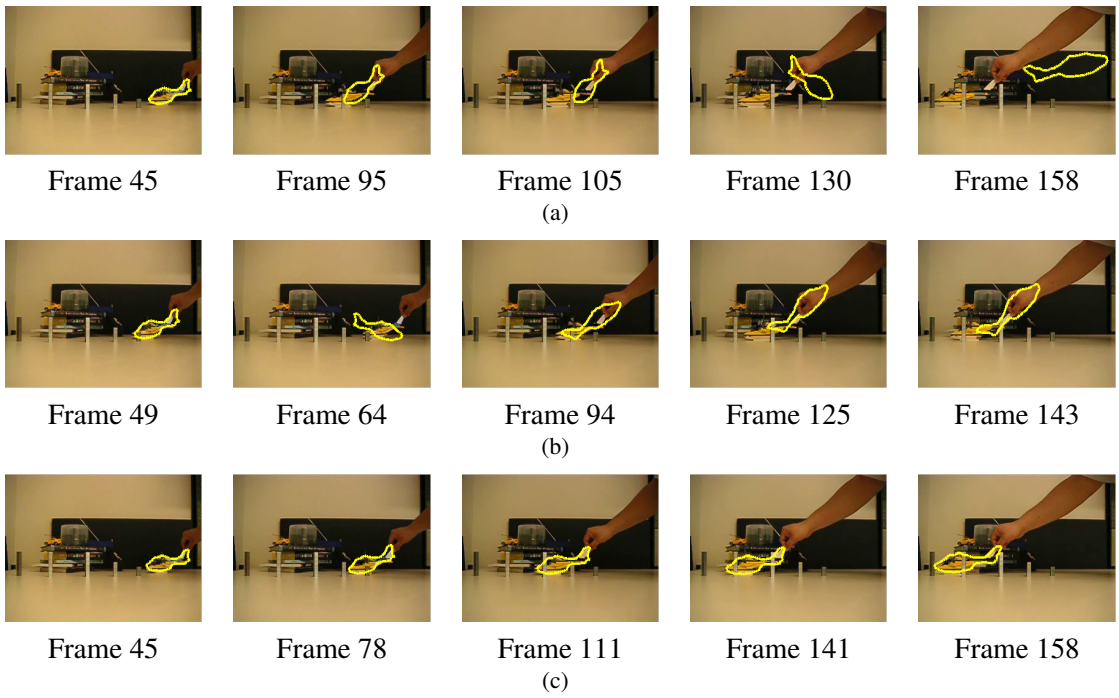


Figure 34: Helicopter sequence I with occlusion in a cluttered environment. Tracking results (a) using the method in [25], (b) using the method in [60], and (c) using the proposed method.

separately distributed samples of pose parameters and the embedded variational technique aid the tracker in finding the optimum in a filtering distribution.

4.9 Chapter Conclusion

In this chapter, we proposed a robust algorithm for 2D-3D pose estimation. We firstly presented a method for 2D-3D visual pose tracking via Monte Carlo sampling on a special Euclidean group SE(3). The use of knowledge of a 3D model allows the tracker to capture the quasi-deformations of an object more accurately than 2D-based trackers. The proposed tracking framework shows promising results in optimally tracking the object of interest in 2D and estimating its pose in 3D under a cluttered environment. However, to effectively handle severe occlusions and to exploit the more natural a particle filtering scheme, we proposed a robust algorithm for 2D-3D pose estimation using a particle filtering approach in conjunction with the variational technique presented by authors in [25]. In the proposed algorithm, the degree of *trust* between predictions and measurements of the system is dynamically controlled in an online fashion as opposed to similar particle filtering algorithms [88, 95]. The resulting methodology was shown to improve tracking performance in continuously locating the target even in the presence of noise, clutter, and occlusions during tracking. In particular, the proposed method shows reliable occlusion handling regardless of significant occlusions that are statistically different from the object of interest with respect to the methods of [25] and [60].

The proposed method has some limitations that we intend to investigate in our future work. First of all, the overall algorithm is computationally expensive despite the benefits described in Section 4.6.4. In addition, our approach could lose the track for non-rigid objects. One possible solution is to incorporate knowledge of multiple 3D shapes as shown in [96]. In other words, for a successful non-rigid tracking framework one should include not only filtering of the pose parameters, but also the shape parameters. This approach would hopefully allow one to track a non-rigid object of interest in some important scenarios.

CHAPTER V

REAL-TIME OBJECT DETECTION USING ACTIVE CONTOURS

In this chapter, we present an algorithm for real-time contour tracking based on fast level set methods. Active contours are usually implemented by level set methods [78, 102]. However, they are not available for real-time applications; active contours implemented in the level set framework produce significant computational burden since the curve is implicitly represented by a higher dimensional function. To speed up level set based curve evolution for real-time applications, we employ the two fast level set algorithms proposed by Song and Chan [107], and Shi and Karl [105]. And, we combine them in the framework of the Chan-Vese active contour model [13]; in this work, we chose region-based active contours of [13] because they are less sensitive to noise and robust to detect an object of interest in cluttered environments rather than edge-based active contours of [52]. Unlike active contours introduced in the previous chapters, this approach provides fast speed to segment and detect an object in a real-time image sequence. The proposed method is designed for real-time detection of multiple windows to help an unmanned aerial vehicle (UAV) automatically locate an entry of a specific building to carry out missions in the 2008 International Aerial Robotics Competition (IARC).

The remainder of this chapter is organized as follows. In the next section, we introduce motivation of this work and the related literature. In Section 5.2, we give an overview of a fast level set implementation of geometric active contours. Next, we propose an approach for real-time detection of multiple windows in Section 5.3. First, we formulate a problem of window detection and we analyze geometric characteristics of a window. Then, methods for shape classification and feature extraction are presented to distinguish windows from noisy structures. We demonstrate experimental results of the proposed method on actual video sequences in Section 5.4. Lastly, we conclude this chapter and discuss possible future research directions in Section 5.5. Much of this chapter is based on the part of [91].

5.1 Introduction and Related Work

Active contours [52] are quite useful tools in performing image segmentation. It is based on the deformation of a contour to capture an object of interest realized as the minimum of an energy functional as described in Chapter 2. Active contours are many times implemented via level set methods [78, 102]. However, the implementation of active contours via level sets is computationally expensive since the higher dimensional function and the computation of the resulting partial differential equations (PDEs) may require higher computational costs. This has limited their availability for real-time applications. Thus, to reduce computational costs, narrow-band level set methods [2, 116] or fast marching methods [102] are generally used for curve evolution because of their fast speed and unidirectional flows. In their methods, the fast speed is achieved by reducing the domain of computation to a narrow bands around the zero level set. While greatly reducing directional complexity, for our special purposes we have found difficulties with these methods due to certain problems with high-gradients edges, which are sensitive to local minima [38, 47]. Moreover, reinitialization and step size control of such narrow band approaches still require high computational cost. To overcome these problems, efficient implementation methods for the Chan-Vese model [107, 105, 79] are proposed to reduce the computational load of curve evolution, in which the speed up of curve evolution is achieved by not solving partial differential equations. The authors of [69] present a fast approximate level set framework, in which the signed distance function is simply represented by integral values and it is required for curve evolution to only compare a few simple integer. In the work of this chapter, we combine the two fast methods of [105] and [107] to achieve fast curve evolution for our UAV flight tests.

5.2 Fast Level Set Implementation of Geometric Active Contours

In this section, we briefly review the two fast level set implementation of active contours and combine them in the framework of the Chan-Vese active contour model. Shi and Karl [105] propose a fast implementation of the level set method at a pixel-wise resolution without solving PDEs. The basic idea of this method is that the curve evolution is only carried out by simply switching the

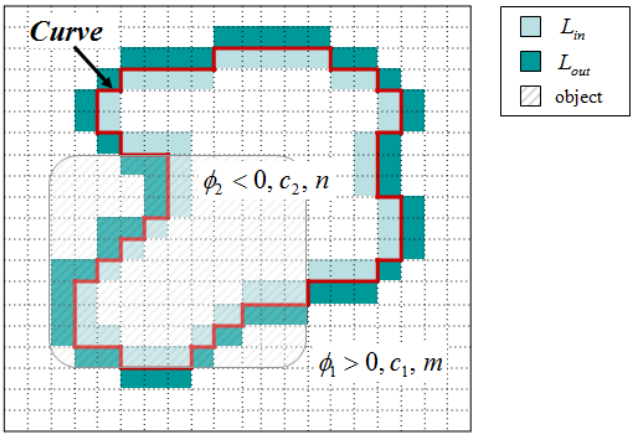


Figure 35: The implicit representation of a curve with the associated two lists and notations inside and outside curve.

neighboring pixels between two lists defined as follows:

$$L_{out} = \{\underline{x} | \phi(\underline{x}) > 0 \text{ and } \exists \underline{y} \in N_4(\underline{x}) \text{ s.t. } \phi(\underline{y}) < 0\}$$
$$L_{in} = \{\underline{x} | \phi(\underline{x}) < 0 \text{ and } \exists \underline{y} \in N_4(\underline{x}) \text{ s.t. } \phi(\underline{y}) > 0\} \tag{73}$$

where $N_4(\underline{x})$ is a 4-connected discrete neighborhood of a pixel $\underline{x}$ and the level set function ϕ is defined as (see Figure 35):

$$\phi(\underline{x}) = \begin{cases} +3 & \text{for} \quad \underline{x} \text{ is an exterior point,} \\ +1 & \text{for} \quad \underline{x} \in L_{out}, \\ -1 & \text{for} \quad \underline{x} \in L_{in}, \\ -3 & \text{for} \quad \underline{x} \text{ is an interior point.} \end{cases} \tag{74}$$

Points that are not in L_{in} and L_{out} are defined as exterior points if $\phi(\underline{x}) > 0$, or interior points $\phi(\underline{x}) < 0$, respectively. In addition, two main rules are defined for curve evolution as follows:

- *switch_in*($\underline{x}$):

  (a) Delete $\underline{x}$ from L_{out} and add it to L_{in}. Then set $\phi(\underline{x}) = -1$.

  (b) $\forall \underline{y} \in N_4(\underline{x})$ if $\phi(\underline{y}) = 3$, add $\underline{y}$ to L_{out} and set $\phi(\underline{y}) = 1$.

- *switch_out*($\underline{x}$):

(a) Delete $\underline{x}$ from L_{in} and add it to L_{out}. Then set $\phi(\underline{x}) = 1$.

(b) $\forall \underline{y} \in N_4(\underline{x})$ if $\phi(\underline{y}) = -3$, add $\underline{y}$ to L_{in} and set $\phi(\underline{y}) = -1$.

The two rules above, *switch_in* and *switch_out*, are related to each evolution step that moves the curve outward and inward by changing the associated pixels, respectively.

Song and Chan [107] present a fast algorithm for variational level set segmentation in a framework of the Chan-Vese model [13]. The main idea of this method is based on observation of segmentation only needing the sign of level set function ϕ, but not its value. The equations for the energy function for curve evolution are as follows:

$$\begin{aligned} \Delta F_{12} &= (\underline{x} - c_2)^2 \frac{n}{n+1} - (\underline{x} - c_1)^2 \frac{m}{m-1} \\ \Delta F_{21} &= (\underline{x} - c_1)^2 \frac{m}{m+1} - (\underline{x} - c_2)^2 \frac{n}{n-1} \end{aligned} \tag{75}$$

where ΔF_{12} and ΔF_{21} denote the differences between the new and old energies when a pixel moves from outside to inside the curve and vice versa, respectively. Here, c_1 and c_2 are the average value of features (intensity values) for the partitioned two regions, ϕ_1 and ϕ_2, respectively. And, m and n are the total number of pixels for ϕ_1 and ϕ_2, respectively, as seen in Figure 35. If the energy decreases after changing a pixel from inside to outside the curve or vice versa, two switch procedures for curve evolution are carried out to satisfy the energy minimization. For example, if $\Delta F_{12} < 0$ when a pixel $\underline{x} \in \phi_1$ changes from ϕ_1 to ϕ_2, then $\underline{x}$ for ϕ_2 is updated to minimize the total energy. By repeating the above procedures until the total energy remains unchanged, the Chan-Vese model is rapidly implemented without explicitly solving any PDE.

The two methods, derived in [105] and [107], complement each other well in the pixel-wise point of view. In addition, their strategies are based on the same concept of switching or changing pixels. Thus, a fundamental idea of the proposed algorithm is to combine two methods by substituting the energy in (75) for the energy function of the fast level set framework. In other words, if $\Delta F_{12} < 0$, then *switch_in* procedure is carried out. If $F_{21} < 0$, then *switch_out* procedure is executed. The curve evolves inward and outward by scanning two lists alternately. The proposed algorithm is described as follows:

(a) Initialize level set function ϕ, two average values c_1 and c_2, the number of pixels m and n, two lists L_{in} and L_{out} from the initial curve.

(b) For all elements of L_{out}, calculate ΔF_{12}. If $\Delta F_{12} < 0$, then do *switch_in*($\underline{x}$) and update c_1, c_2, m, and n:

$$c_1 = c_1 + \frac{c_1 - \underline{x}}{m-1}, \quad c_2 = c_2 - \frac{c_2 - \underline{x}}{n+1} \tag{76}$$
$$m = m - 1, \quad n = n + 1.$$

(c) For all elements of L_{in}, if $\forall \underline{y} \in N_4(\underline{x})$ and $\phi(\underline{y}) < 0$, then delete $\underline{x}$ from L_{in} and set $\phi(\underline{x}) = -3$.

(d) For all elements of L_{in}, calculate ΔF_{21}. If $\Delta F_{21} < 0$, then do *switch_out*($\underline{x}$) and update c_1, c_2, m, and n:

$$c_1 = c_1 - \frac{c_1 - \underline{x}}{m+1}, \quad c_2 = c_2 + \frac{c_2 - \underline{x}}{n-1} \tag{77}$$
$$m = m + 1, \quad n = n - 1.$$

(e) For all elements of L_{out}, if $\forall \underline{y} \in N_4(\underline{x})$ and $\phi(\underline{y}) > 0$, then delete $\underline{x}$ from L_{out} and set $\phi(\underline{x}) = 3$.

(f) Check the stopping condition, i.e., if $\Delta F_{12} > 0$ for all elements of L_{out} and $\Delta F_{21} < 0$ for all elements of L_{in}, then terminate the procedure, otherwise go to step (b).

In the proposed algorithm above, the computational cost is drastically reduced because curve evolution via level set methods is achieved without solving any PDE. In addition, boundary information of the curve is preserved in the two lists so that it is used for contour-based shape analysis to acquire a specific target, which will be described in the next section.

5.3 Application to Real-time Detection of Multiple Windows

Fast curve evolution proposed in Section 5.2 is applied to detect multiple windows in sequential images from a single camera mounted on an unmanned aerial vehicle (UAV); it is designed for UAV to carry out a mission finding openings, i.e., windows, of a specific building in the 2008 International Aerial Robotics Competition (IARC). Target detection for dynamic image data can be achieved through sequential segmentation over image frames. Some feature points, such as corner points and the centroid of windows, are used for calculating the global location of the windows in a 3D space.

5.3.1 Geometric Characteristics for Window Detection

For window detection, we take advantage of geometric characteristics of a window as follows:

- The intensity value of the window is less than a background. The color of the window is usually dark.
- The shape of the window is rectangular or parallelogram satisfying symmetry with a four-sided shape.
- Shadows are bigger than the window or they are not rectangular in shape. The boundary line of noise is not smooth.

The intensity values of all pixels of the given image I are normalized to lie between 0 and 1 as

$$I = \frac{I - \min(I)}{\max(I) - \min(I)}. \tag{78}$$

Only regions including a window or windows are selected for the next segmentation process. To this end, the given image is divided into small spatial regions and a checking process is executed to find regions satisfying the following condition:

$$N^p(I(\overrightarrow{k}) < I_{th}) > N^p_{th} \tag{79}$$

where $\overrightarrow{k}$ is a pixel vector and $N^p(\cdot)$ is a function that indicates the total number of pixels satisfying the given certain condition. I_{th} and N^p_{th} are an intensity threshold and the threshold of the number of pixels, respectively. If a certain region satisfies condition (79) (i.e., if the total number of pixels whose intensity are less than I_{th} is greater than the pre-defined N^p_{th}), the proposed curve evolution algorithm is carried out. The computational time for segmentation is drastically reduced through the above pre-processing.

5.3.2 Shape Analysis and Feature Extraction

After completing the segmentation process, several segmented contours become candidates as the most likely windows. Since our images include unwanted objects whose shape is similar to a window, such as shadows, grass, mud, etc., shape analysis is an important process to filter out such unexpected objects among them. First of all, we extract the connected set of boundary pixels of

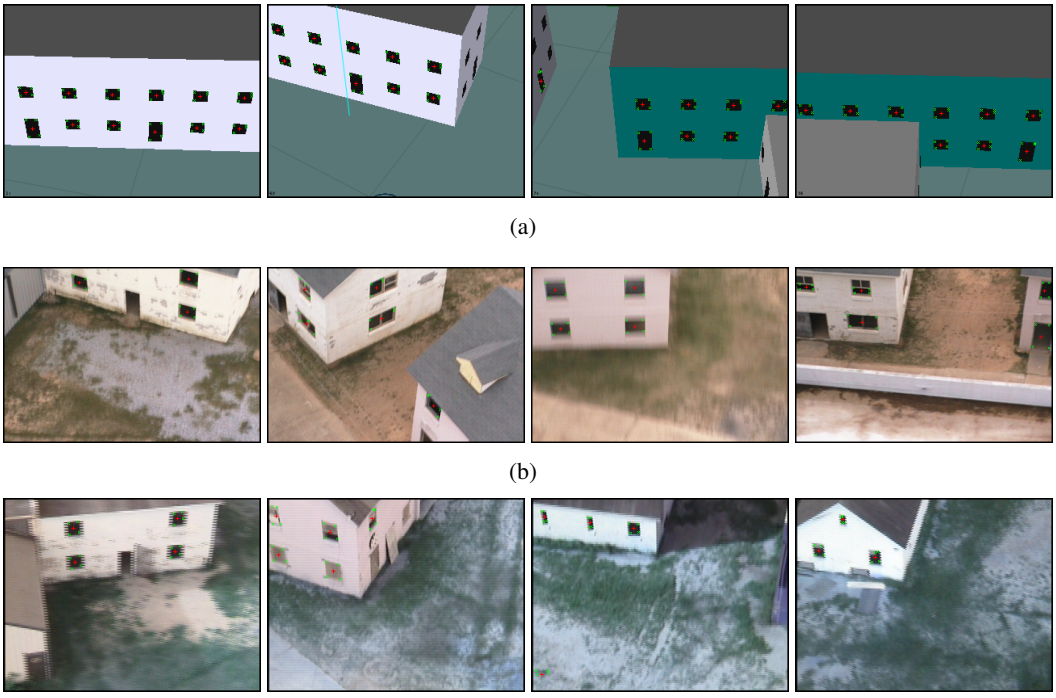
the selected candidate [91]. Since the lists provide information of the pixels defining the discrete version of the closed contour, the connected components of each contour can be extracted by tracing around the pixels in the lists. While doing this, we obtain the histogram of the orientation difference between previous and current pixels; it provides geometric characteristics of a curve, such as a vertical-to-height ratio and a degree of symmetry. Now, we filter the unexpected objects out by measuring the degree of similarity to the window according to some criteria mentioned in Section 5.3.1. This method is simple and it does not require high computational cost though it provides enough information for shape analysis for our tasks.

For the same purpose, one can use the methods described in [21] to measure degrees of rectangularity and symmetry. For example, rectangularity can be defined as how shape can be approximated by a minimum enclosing rectangle: $A_{\mathbf{win}}/A_{\mathbf{mer}}$ where $A_{\mathbf{win}} = \int_{\phi(\underline{x})\in -1,-3} d\underline{x}$ and $A_{\mathbf{mer}}$ are the area of the window and the minimum enclosing rectangle, respectively. Here, the minimum enclosing rectangle is defined as the smallest rectangle enclosing the shape, which may not be oriented with respect to coordinate axes. For the degree of symmetry, bilateral symmetry is estimated from binary image representations. Its degree can be estimated by computing a difference between original shape and its reflected superimposition. Reflection is carried out with respect to the line having orientation of the first principal component or a major axis of shape.

After the completion of window detection, all of geometric information of the detected windows, such as corner locations, areas, intensity value of the interior, and centroid, are calculated and transferred to UAV system to estimate the location of windows in a 3D space. In this work, the corner locations of windows are obtained by using information given by the change of orientation between the connected points of a closed contour. The basic idea of this is to check how much orientation changes while tracing around boundary points of a closed contour until returning to the starting point. If the accumulated sum of radians until a current pixel is greater than $\pi/2$, the pixel becomes a corner point.

5.4 Experiments

The window detector was tested on both synthetic images from a simulation and real images from IARC using a Pentium4 3.6GHz, 2GB PC. The results of the proposed algorithm are shown in



(a)

(b)

(c)

Figure 36: Successful results of the proposed scheme for (a) a synthetic sequence from a simulation program, and (b) and (c) outdoor sequences from IARC. Frame order: from left to right.

Figure 36. The algorithm detected multiple windows quickly and robustly, and it found their center and corner points in real-time speed. The corner points and center point are marked by x-marks and cross-hairs, respectively. The average processing time takes 0.0049sec/frame, which is enough speed to track multiple windows in real-time sequences.

5.5 Chapter Conclusion

Level set methods provide useful properties to automatically deal with topological changes of curve evolution. However it suffers from high computational cost that leads to limitation of a practical use for real-time applications. Thus, to reduce the computational complexity in the level set framework, we introduced the fast algorithm for level set based optimization proposed by Song and Chan [107] and the fast pixel level curve evolution proposed by Shi and Karl [105]. Then, we combine the two fast algorithms in the framework of Chan-Vese active contours. The proposed algorithm provides enough speed and successful results to track multiple windows in real-time image sequences as shown in our experiments in Section 5.4.

However, the proposed window detector has limitations for some cases. For example, if the UAV rapidly approaches a building, the proposed method may not detect a window because its size and intensity value can dramatically change. In addition, if the shape of a shadow is similar to a window, it is necessary to apply additional criteria to distinguish each other, such as geometric location or the number of windows of the targeted building. In addition, the mean intensity separation model of the Chan-Vese active contours used in this work often leads to incorrect segmentation because it assumes that distributions are of unit variance. Thus, one can change or extend the energy model according to the degree of image noise. For example, a Gaussian model or generalized histograms could be considerable for more robust segmentation.

CHAPTER VI

HUMAN BODY TRACKING AND JOINT ANGLE ESTIMATION FROM MOBILE-PHONE VIDEO FOR CLINICAL ANALYSIS

In this chapter, we deal with rapid human motion in the context of object segmentation and visual tracking. Specifically, we propose an algorithm for tracking human jumping motion and estimating a joint angle of the knees for clinical analysis. Compared to general object tracking, tracking for rapid human motion, such as jumping, dodging, and running, is not characterized by linear dynamic models (e.g., a constant velocity model or a constant acceleration model). In this case, an approach without a pre-defined motion model could be a more feasible solution. For example, the background modeling and subtraction, or template learning and matching are methods based on iterative detection without using a motion model of an object. These methods can be exploited for tracking by merging object detection of each frame in a sequence. In this chapter, we use an adaptive pixel-color model to effectively detect an object of interest throughout a sequence; it allows the proposed tracker to capture a time-varying distribution of appearance information of an object. More specifically, we segment interesting parts of a human body at each frame in the given sequence instead of the use of a specific motion model, which is usually adopted in filtering based methods described in the previous chapters. Moreover, we propose a model-free and marker-less approach for human body tracking, which is applicable to a daily environment where no professional medical facilities are available.

The remainder of this chapter is organized as follows. In the next section, we introduce the work of this chapter and provide some related works to the proposed approach. In Section 6.2, we describe the proposed algorithms for human body tracking and joint angle estimation. First, a method for body part segmentation and partition is described. Next, eigen-axes based joint angle estimation for clinical analysis is presented. In Section 6.3, the proposed algorithm was tested on sequences taken from a consumer-level mobile phone. Lastly, we conclude this chapter and discuss possible future research directions in Section 6.4. Much of this chapter is based on [58].

6.1 Introduction and Related Work

Tracking a human body's joint angle enables important clinical analysis for post-operative analysis and prediction of a healthy subject's likelihood of injury [46]. To enable such video-based estimation in a minimally controlled environment with consumer-grade hardware, we present a model-free and marker-less joint angle tracking and estimation method. By restricting the tracking task to the body segments of interest and employing a dynamic online estimate of the color distribution, fast and robust tracking results are obtained from a low signal-to-noise-ratio monocular camera source.

A vast number of tracking methods for a human body are proposed in literature [74]. Most commonly, they adopt a explicit body model, such as a model built on a kinematic chain consisting of conical sections [26] or a model composed of a set of kinematic and tapered cylinder shape [62]. To optimize the map of image-features to model, the work in [26] introduced annealed particle filters suitable for high dimensional configuration spaces. In [62], data-driven MCMC technique is proposed to estimate 3D poses showing the best model match. In contrast, our method is carried out without explicitly using a body model and markers that can indicate and describe a human figure. The motivating use-case for our proposed algorithm is a hand-held camera of a mobile phone in a home or office setting. Accordingly, only a monocular camera can be used; approaches using multiple cameras [10] are outside the scope of feasibility in common daily environments.

Noise and clutter in such an uncontrolled setting further complicates the already non-trivial task of human body segmentation. Region-based active contours are a prevalent method for image segmentation in the presence of noise [73]; they are generally formulated to evolve by gradient descent. Methods based on active contour models show a robust result in segmenting a region of interest from noisy backgrounds. While this approach will in many cases robustly segment a desired region of interest from a cluttered background, it is highly sensitive to initialization and can easily become stuck in a local minimum. In addition, this framework is not suitable in tracking a human body with fast and blurred motion and dealing with the disappearance of some parts of a human body due to narrow field-of-view of a phone camera. Therefore, we employ active contour models for initialization of our tracking framework. Then, a dynamic color-model based segmentation is proposed to properly segment a human body and maintain the track.

Segmenting objects accurately and reliably requires a robust color model. Modeling the skin-color is a challenge that must be addressed in marker-less human body segmentation: numerous background objects, such as a wooden desk or white wall, in tandem with indoor lighting, make skin regions very similar in pixel-value to portions of the background. During a video sequence, the subject's motion can dramatically change the apparent pixel-values due to overhead lights being obscured by their arms, head, and torso; even for a specific human and environment, the observed color values can fluctuate significantly. Most methods for the construction of a skin-color model that robustly characterizes human skin under varying lighting conditions and skin tones rely on supervised training with several images before segmentation begins [85]. Naturally such an approach has limitations: collecting and training with example images of skin is a time-consuming task and cannot completely cover the possible range of a skin-color distribution. We propose an online adaptive pixel-color model which captures the time-varying distributions of skin and clothing color.

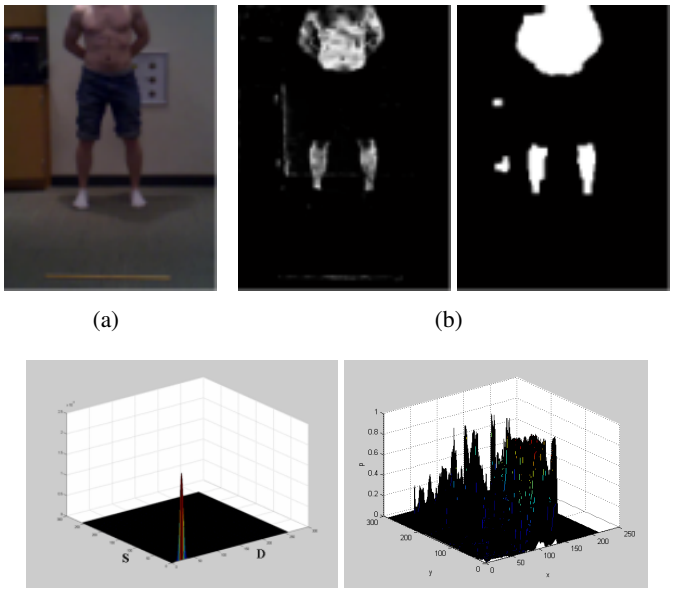
6.2 Proposed Algorithm

6.2.1 Human Body Segmentation and Partitioning

The RGB (Red, Green, and Blue) representation of color images is most common but is not suitable for characterizing a skin-color region because the chrominance and luminance information are correlated in the RGB space. Luminance has an effect on skin color according to the varying levels of brightness. To overcome this, the robustness of several color spaces is analyzed in [101]. For example, normalized RGB showing chromatic colors is effectively used for skin color segmentation [117]. In this present work, the combination space of a difference space (D) between a red space (R) and a green space (G), and a saturation space (S) of a color image is used. This space provides a good indicator for some colors. In particular, skin color is well classified and its distribution resides within $[15, 90]$ in a D space via our empirical observations. We denote by I the image, by $I_{(\cdot)}$ the associated component (or space) of an image I. The image used in this work is defined as:

$$\begin{aligned} I_D &= I_R - I_G \\ I_S &= \frac{\max(I_R, I_G, I_B) - \min(I_R, I_G, I_B)}{\max(I_R, I_G, I_B)}. \end{aligned} \tag{80}$$

The proposed color model is statistically designed as a multivariate Gaussian distribution. The likelihood of color for a pixel, $\mathrm{x} \in \mathbb{R}^2$, with a pair value of $\mathrm{m} = [d, s]^T$, $d \in I_D$, $s \in I_S$, is



(a) (b)

(c)

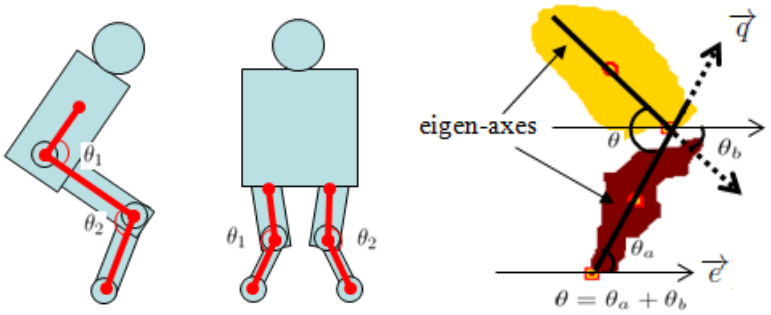
Figure 37: Skin segmentation: (a) An original image. (b) A result using the proposed method (left) and its smoothed result by a morphological filter (right). (c) A skin color model (left) and the likelihood of skin for the image (right).

obtained from the color model, which is given by:

$$p(\mathrm{x}|\mathrm{m}) = |\Sigma_c|^{-\frac{1}{2}} \exp -\frac{1}{2}(\mathrm{m}-\mu)^T {\Sigma_c}^{-1}(\mathrm{m}-\mu) \tag{81}$$

where $\mu \in \mathbb{R}^2$ is a mean value vector for D and S spaces and $\Sigma_c \in \mathbb{R}^{2\times 2}$ is a covariance matrix. The pixels from (81) are labeled as a foreground image of interest if they satisfy $p(\mathrm{x}|\mathrm{m}) > c_{th}$. Here c_{th} is a user-defined constant for the threshold and is empirically selected as $[0.3, 0.5]$ in our experiments. Therefore, it produces an binary image, $b(\mathrm{x})$, representing the segmented region as 1 and backgrounds as 0. The binary image is smoothed by a morphological filter, $f_s(\cdot)$: $\bar{b}(\mathrm{x}) = b(\mathrm{x}) \otimes f_s(\cdot)$. The size of the filter depends on the noise present in the sequence. This allows improved segmentation results in a highly cluttered environment. Since $\bar{b}(\mathrm{x})$ can be composed of several pixel groups, for convenient notation, we let $g^i(\mathrm{x})$ be each labeled pixel group and its region is denoted by R^i. Figure 37 shows segmentation results of skin color by using the proposed method.

After segmentation based on the color model, the labeled regions are selected as each body part of interest among several labeled regions by considering the centroid and the size of the previously



(a) (b)

Figure 38: (a) Joint angles of interest at a side view (left) and a front view (right). (b) Joint angle estimation using eigen-axes

segmented body parts as follows:

$$E_c^{\star} = arg \min_{g^i(\mathrm{x})} E_c^i \tag{82}$$

where

$$E_c^i = \|(c_{\mathrm{x}}^i - c_{\mathrm{x}}(0))^2 + \left(\int_{\mathrm{x} \in g^i} \mathrm{x} d\mathrm{x} - \int_{\mathrm{x} \in g(0)} \mathrm{x} d\mathrm{x} \right)^2 \|. \tag{83}$$

Here $\| \cdot \|$ is the Euclidean norm and $c_{\mathrm{x}}^i \in \mathbb{R}^2$ is the centroid of each labeled region R^i at a current frame. $c_{\mathrm{x}}(0)$ and $g(0)$ is a centroid and a labeled group of the previously selected region, respectively. The centroid of a labeled region is computed by the average value of the points' coordinates inside the region R^i [21]. A labeled region with the minimum energy E_c is selected as a region of interest, such as torso, upper and lower legs.

6.2.2 Joint Analysis

The segmented human body can provide a human motion analyst with crucial information, such as the joint angle between two pieces of limbs. The proposed method is designed to help an orthopedist study healthy subjects and see the probability of injury based on posture of jumping and landing and to study recovery in post-operative patients. To this end, joint angles of a human jumping and landing are estimated in a daily environment where professional medical facilities are not available. Some joint angles of interest in this work are shown in Figure 38 (a); angles between knees and toes, and angles between a knee and a hip. Some minimal control of the experimental setup is assumed. First, the human is instructed to jump either along a line in the camera plane *side-view* or facing the lens *front-view*. With the narrow angle-of-view in the mobile-phone camera source, lens distortion

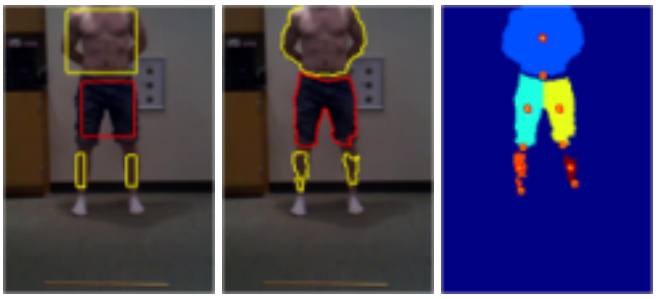


Figure 39: Initialization process of tracking: (from left to right) initial contours, final contours, and body part separation and interesting point detection, i.e., circle points and square points indicate the centroid of each body part and joint points of interest, respectively.

is negligible. The individual is wearing some specific clothing (*e.g.* homogenously colored pants). There is no constraint on a specific color or reflective markers.

In this work, eigen-axes are used for estimating joint angles between body parts because they provide orientation information of the shape derived from a binary pixel group g^i. Eigen-axes are defined as the eigenvectors of a covariance matrix of all vectors representing the point coordinates inside R^i [21]. Figure 38 (b) illustrates the method of calculating the angle between a upper leg and a lower leg using a major principle axis. In Figure 38 (b), the angle is simply calculated by using trigonometric functions: $\theta_{(\cdot)} = \cos^{-1}(\overrightarrow{q} \cdot \overrightarrow{e}) \times \frac{180}{\pi}$ where $\overrightarrow{q}$ and $\overrightarrow{e}$ are a major principle axis vector and a horizontal axis vector, respectively. A joint position is obtained from determining the intersection point of boundaries of g^i, the line defined with inclination $\theta_{(\cdot)}$, and a centroid c^i_{x}.

6.2.3 Human Motion Tracking

Initialization to extract the color information of a human body that will be tracked is manually achieved at the first frame of a sequence. To this end, one can use any type of segmentation methods, such as region growing or active contour models. In our work, we use a region-based active contours driven by Bhattacharyya gradient flow described in [73]. To use this, it is assumed that the human body of interest is fully viewed within a camera view angle at the first frame. To generate the color models for each color, initial contours are carefully located so that they evolve over objective color regions. Figure 39 shows an initialization process at the first frame of a sequence shown in Figure 41. Here two color models are generated. They are re-generated online for each frame during a sequence to capture the time-varying distributions of colors, which is based on the result of the

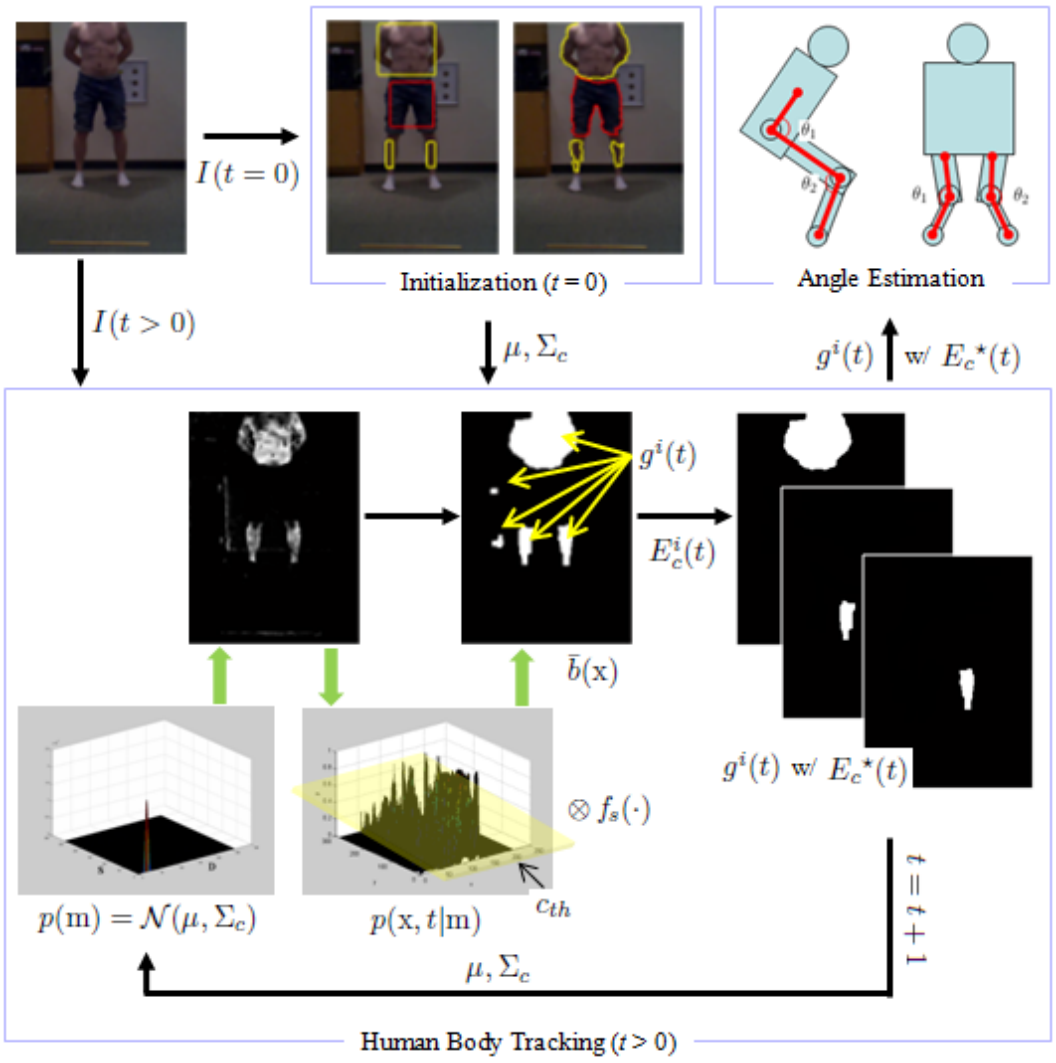


Figure 40: Schema summarizing the proposed human body tracking and joint angle estimation.

previous frame. The proposed tracking framework is described as follows:

(a) Initialize active contours and evolve them to define regions of interest at $t = 0$ where t is a time step.

(b) Generate color models for each color region and filter the image from (81): $p(\mathrm{x}, t = 0|\mathrm{m})$.

(c) Segment and label a pixel group: $g^i(t)$.

(d) Compute the energy E_c^i for each labeled region g^i by using (83) to decompose the segmented

human body into each body part of interest:

$$E_c^i(t) = \|(c_{\mathrm{x}}^i(t) - c_{\mathrm{x}}(t-1))^2 + \left(\int_{\mathrm{x} \in g^i(t)} \mathrm{x} d\mathrm{x} - \int_{\mathrm{x} \in g(t-1)} \mathrm{x} d\mathrm{x} \right)^2 \|.$$

(e) Update color models based on the previously selected regions of each body part $g^i(t-1)$: $p(\mathrm{x}, t|\mathrm{m})$.

(f) Go to step c: $t = t + 1$.

The schema of the proposed tracking framework is shown in Figure 40.

6.3 Experiments

The proposed algorithm was tested on sequences of human jumping and landing showing front and side views of human motion in real environments. All sequences are taken from a low-resolution, small-aperture, narrow field-of-view monocular camera built into a mobile phone in cluttered backgrounds. Due to these properties of the used mobile-phone camera, the acquired images have high noise and low quality. In addition, some parts of the tracked human body (in particular, the torso) are out of camera view in some frames due to a narrow angle lens.

Figures 41 and 42 show the experimental results of the proposed algorithms for front and side views of fast jumping and landing sequences, respectively. Acceptable segmentation and tracking results are obtained and joint angles of interest are estimated to analyze jumping and landing posture. Note that the quality of images is poor and the tested sequences show fast motion. To highlight the effectiveness of the proposed dynamic color model in capturing the time-varying color distributions, results by using a static color model acquired at the first frame are also shown in Figure 41. Figure 43 displays the graphs of joint angles of interest, θ_1 and θ_2, over the sequences shown in Figure 41 and 42, respectively.

6.4 Chapter Conclusion

In this chapter, we have shown an effective algorithm to track a human body and estimate its joint angles by incorporating dynamic color-model based segmentation and eigen-axis based angle estimation. No explicit human model and markers is needed in the proposed framework. The results

of experiments showed robust performance of the proposed algorithm and applicability of a joint angle analysis.

However, the proposed algorithm has some limitations that we intend to overcome in our future work. Since the proposed algorithm largely depends on the segmented body parts to extract their axis points and estimate joint angles, it is difficult to guarantee the accuracy of joint angles due to missing or inaccurate segmentation. Thus, one can employ a 2D or 3D body model to complement poor segmentation for the improvement of angle estimation.

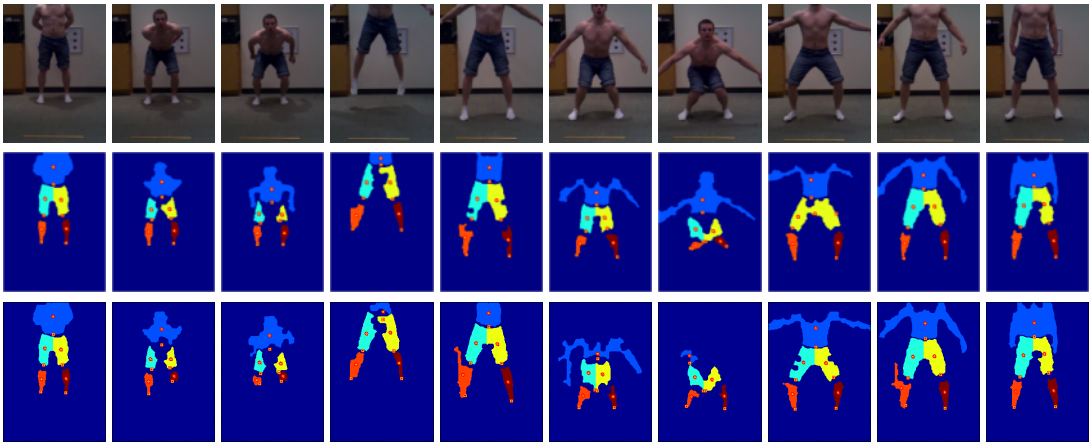


Figure 41: Sequence I: *front-view* jumping and landing posture. The torso partially disappears while jumping as a consequence of narrow angle-of-view. Incident overhead lighting greatly varies during the sequence. Bottom row: results using a static color model.

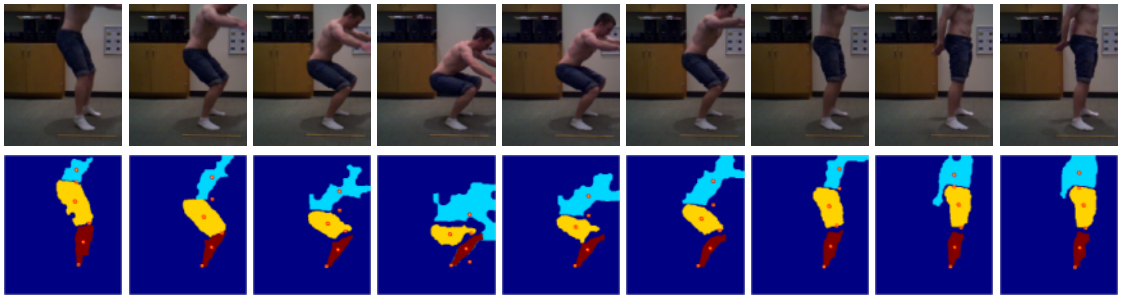


Figure 42: Sequence II: *side-view* jumping and landing posture. There is a disappearance of arms and head from the frame during the sequence. The dynamic color model adapts to observed pixel values.

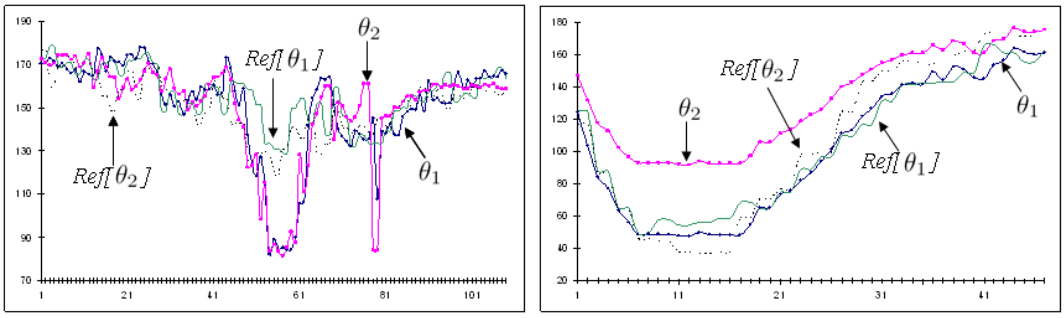


Figure 43: Graphs of estimated angles of the sequences of Figure 41 (top) and the sequences of Figure 42 (bottom). A circled line and a squared line denote the angle of θ_1 and θ_2 of each viewpoint, respectively. Their reference angles are denoted by a solid line and a dotted line, respectively. Extrema of the angles θ_1 and θ_2 encode *maximal knee flexion* [46].

CHAPTER VII

CONCLUSION

In this thesis, we proposed novel algorithms for object tracking and pose estimation. All tracking frameworks proposed in this thesis are based on or inspired by variational methods such as geometric active contours and the Bayesian inference with the Monte Carlo methods such as particle filters. Moreover, several techniques are adopted to deal with challenging and real tracking scenarios to robustly track the object and to continuously maintain its track. In particular, the proposed algorithms were designed to show the potential in handling partial occlusions and target reacquisition during tracking by incorporating 2D or 3D shape information of the object and exploiting the systematic control of the particle filters in the context of visual tracking. Additionally, we introduced the fast level-set based algorithm applicable to real-time applications in which the contour-based tracker using the level set methods was improved in terms of computational complexity. We also presented a model-free and marker-less method using a dynamic color model for tracking the human body with rapid motion; in this case, a dynamics for the motion is not available. The contributions of this thesis are summarized below.

In Chapter 3, we proposed a reliable algorithm to track a deformable object in a time-varying sequence of 3D range data by combining particle filtering and geometric active contours. In this algorithm, the depth maps are statically or dynamically reconstructed to improve the segmentation process of the region-based active contours. In addition, this work presented an on-line shape learning and matching method based on PCA to reacquire track of an object in the event that it disappears from the field of view and reappears later.

In Chapter 4, we proposed an approach to jointly track the location of a rigid object in 2D and to estimate its pose in 3D from a 2D image sequence. First, we proposed a Monte Carlo based sampling method to estimate a 3D transformation matrix for 2D-3D pose tracking. In addition, we used prior knowledge of a 3D model of an object to improve the tracking performance. Next, we employed particle filtering to generate and propagate the translation and rotation parameters in a

decoupled manner. Moreover, we developed a scheme for occlusion detection and handling based on controlling of the degree of dependencies between predictions and measurements of the tracking system. This scheme allows the tracker to continuously track the object in the presence of severe occlusions. For this scenario, the region-based energy is usually not appropriate.

In Chapter 5, we proposed a fast level set based algorithm applicable to real-time applications. To reduce the computational complexity of curve evolution in the level set framework, we introduced the fast algorithm for level set based optimization, and the fast level set based on pixel level curve evolution. Then, we combined the two fast algorithms in the framework of the Chan-Vese active contour model. This tracker successfully detected the multiple windows in a real-time image sequence.

In Chapter 6, we showed an effective algorithm to track a human body and estimate its joint angles by incorporating dynamic color-model based segmentation and eigen-axis based angle estimation. Since the proposed framework was independent of specific motion models, explicit human models, and markers, it is simple and applicable for processing videos from consumer-level cameras. The results of the experiments showed robust performance of the proposed algorithm for joint angle analysis in a clinical setting.

Most of the novel frameworks proposed in this thesis were applied to object tracking and pose estimation. However, besides suggestions mentioned in each chapter, they could be further applied to various applications in computer vision. For example, the proposed algorithms could be used for the development of human gesture recognition. This system often uses a depth-aware camera to generate a depth map so that it can track a 3D pose of the face or hand. However, the face or hand easily disappears from the field of view due to the narrow angle-of-view of the camera. Thus, the weighting scheme of stereo data and target reacquisition method proposed in Chapter 3 can be effectively applied to the system for robust tracking of the face or hand. Another application for further research would include an object retrieval and classification system in the context of image and video search. In this case, using a 3D shape prior and 2D-3D pose estimation introduced in Chapter 4 can be useful because an object appears in various poses in an image. In addition, this system can be extended to a search system for a non-rigid object by exploiting multiple 3D shapes of the object to capture its wide range of deformations.

Many of researchers have made every efforts and are still studying to understand a human visual system and to design an artificial vision system similar to it. We believe that visual tracking is one of the most important and fundamental technologies in the field of computer vision and it could naturally allow us to better understand the human visual system. Finally, we conclude this thesis as hoping that it has made a contribution (even though it is small) towards the research on computer vision and visual tracking.

APPENDIX A

DERIVATIONS OF THE GRADIENT FLOW FOR REGION-BASED ACTIVE CONTOURS

In this appendix, detailed computations of the gradient flow for region-based active contours, which is introduced in Chapter 2, are presented. Let an energy that minimizes a particular function f inside a region R enclosed inside a curve C be as

$$E(C) = \int_R f(\mathbf{x}) d\mathbf{x} \tag{84}$$

where $\mathbf{x} = (x,\ y)$. Given $\mathbf{F}(\mathbf{x})$ that is a vector field chosen so that $\nabla \cdot \mathbf{F}(\mathbf{x}) = f(\mathbf{x})$, from the divergence theorem, the region integral based on f is equal to the contour integral based on $\mathbf{F}$. Thus, (84) becomes

$$E(C) = \oint_C < \mathbf{F},\ \mathbf{N} > ds \tag{85}$$

where $\mathbf{N}$ denotes the unit normal of C, ds is the Euclidean arc length element.

First, we rewrite (85) in terms of a fixed parameterization $p \in [0,\ 1]$ of the curve C, which is independent of time t as the curve evolves, as follows:

$$\begin{aligned} E(C) &= \int_0^1 < \mathbf{F},\ \frac{JC_p}{\|C_p\|} > \|C_p\|\, dp \\ &= \int_0^1 < \mathbf{F},\ JC_p > dp \end{aligned} \tag{86}$$

where the direct $\frac{\pi}{2}$-rotation matrix $J = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$. Here $ds = \|C_p\|\, dp$ and $\mathbf{N} = \frac{JC_p}{\|C_p\|}$. Now, by differentiating E with respect to t, we have

$$\frac{dE}{dt} = \int_0^1 \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_t,\ JC_p \right\rangle + \langle \mathbf{F},\ JC_{pt} \rangle\, dp \tag{87}$$

where $\frac{d\mathbf{F}}{d\mathbf{x}}$ denotes the Jacobian matrix of $\mathbf{F}$ with respect to $\mathbf{x}$, i.e.,

$$\frac{d\mathbf{F}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial F^x}{\partial x} & \frac{\partial F^x}{\partial y} \\ \frac{\partial F^y}{\partial x} & \frac{\partial F^y}{\partial y} \end{bmatrix}. \tag{88}$$

Now, using the integration by parts, the second term of (87) becomes

$$\int_0^1 \langle \mathbf{F},\ JC_{pt} \rangle\, dp = \langle \mathbf{F},\ JC_t \rangle\,|_1^0 - \int_0^1 \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_p,\ JC_t \right\rangle dp = -\int_0^1 \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_p,\ JC_t \right\rangle dp \tag{89}$$

where the first term of (89) is zero because $C(0) = C(1)$ for a parameterized closed curve, and thus $C_t(0) = C_t(1)$. Now (87) becomes

$$\begin{aligned}
\frac{dE}{dt} &= \int_0^1 \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_t,\ JC_p \right\rangle - \left\langle \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_p,\ JC_t \right\rangle dp \\
&= \int_0^1 \left\langle C_t,\ \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right)^T JC_p \right\rangle - \left\langle J^T \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) C_p,\ C_t \right\rangle dp \\
&= \int_0^1 \left\langle C_t,\ \left(\left(\frac{d\mathbf{F}}{d\mathbf{x}}\right)^T J - J^T \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) \right) C_p \right\rangle dp.
\end{aligned} \tag{90}$$

By changing dp into ds and using the fact that $C_s = \mathbf{T}$, $J\mathbf{T} = \mathbf{N}$, and $\mathbf{T} = J^T\mathbf{N}$, we have

$$\begin{aligned}
\frac{dE}{dt} &= \oint_C \left\langle C_t,\ \left(\left(\frac{d\mathbf{F}}{d\mathbf{x}}\right)^T J - J^T \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) \right) C_s \right\rangle ds \\
&= \oint_C \left\langle C_t,\ \left(\left(\frac{d\mathbf{F}}{d\mathbf{x}}\right)^T - J \left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) J^T \right) \mathbf{N} \right\rangle ds \\
&= \oint_C \left\langle C_t,\ \left(\mathbf{tr}\left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) I \right) \mathbf{N} \right\rangle ds
\end{aligned} \tag{91}$$

where $\mathbf{tr}(\cdot)$ and I denote the trace of the given matrix and the unit matrix in linear algebra, respectively.

Finally, since $\mathbf{tr}\left(\frac{d\mathbf{F}}{d\mathbf{x}}\right) I = \nabla \cdot \mathbf{F}(\mathbf{x}) = f(\mathbf{x})$, we obtain

$$\frac{dE}{dt} = \oint_C \langle C_t,\ f\mathbf{N} \rangle\, ds. \tag{92}$$

Thus, the gradient descent flow for C is

$$\frac{\partial C}{\partial t} = -f\mathbf{N}. \tag{93}$$

Note that $\mathbf{N}$ is the outward normal and the flow depends only upon f, not a particular choice for $\mathbf{F}$.

If we define the energy as

$$E(C) = \int_R f_{in}(\mathbf{x}) d\mathbf{x} + \int_{R^c} f_{out}(\mathbf{x}) d\mathbf{x} \tag{94}$$

where f_{in} and f_{out} are for the regions, R and R^c, which are enclosed inside and outside the curve C, respectively. Straightforwardly, one can derive the following gradient descent flow of C from the equations above by the same fashion:

$$\frac{\partial C}{\partial t} = -(f_{in} - f_{out})\mathbf{N}. \tag{95}$$

APPENDIX B

ACTIVE CONTOURS DRIVEN BY THE BHATTACHARYYA GRADIENT FLOW

In this appendix, detailed computations of the Bhattacharyya gradient flow in the level set framework, which is used in Chapter 3, are presented. In level set methods, a closed curve C is represented as the zero level set of a higher dimensional function $\phi : \mathbb{R}^2 \to \mathbb{R}$, which is typically chosen to be a signed distance function such that $\phi < 0$ inside C and $\phi > 0$ outside C. Therefore, the curve can be described by an implicit surface:

$$C = \{\mathbf{x} \mid \phi(\mathbf{x}) \equiv 0,\ \mathbf{x} \in \Omega\} \tag{96}$$

where Ω is the domain of an image $I(\mathbf{x})$: $\mathbb{R}^2 \to \mathbb{Z}$ which is mapping to the photometric variable $z \in \mathbb{Z}$ from the image plane. The Bhattacharyya distance between two probability density functions (pdfs), P_i and P_o, is defined by

$$D_B = -\log(B) \tag{97}$$

where

$$B(\phi) = \int_{\mathbb{Z}} \sqrt{P_i(z)P_o(z)}dz. \tag{98}$$

The Bhattacharyya coefficient B varies between 0 and 1 (0 indicates complete mismatch while 1 represents perfect similarity). Here, P_i and P_o are pdfs defined inside and outside curve C, respectively, in the case of curve evolution, defined by

$$P_i(z) = \frac{\int_\Omega K(z - I(\mathbf{x}))d\mathbf{x}}{\int_\Omega d\mathbf{x}}, \quad P_o(z) = \frac{\int_\Omega K(z - I(\mathbf{x}))d\mathbf{x}}{\int_\Omega d\mathbf{x}} \tag{99}$$

which are the nonparametric kernel density estimates of two pdfs of z for a given kernel K. Popular choices for the kernel K are either Gaussian Kernel or the Dirac delta function. Now, let $H(\cdot)$ be the Heaviside step function defined as

$$H(\phi) = \begin{cases} 1, & \text{for} \quad \phi \geq 0, \\ 0, & \text{for} \quad \phi < 0. \end{cases} \tag{100}$$

Now, we rewrite $P_i(z)$ and $P_o(z)$ in (99) in terms of the level set function ϕ using the Heaviside step function H:

$$P_i(z) = \frac{\int_\Omega K(z - I(\mathbf{x}))H(-\phi(\mathbf{x}))d\mathbf{x}}{\int_\Omega H(-\phi(\mathbf{x}))d\mathbf{x}}, \quad P_o(z) = \frac{\int_\Omega K(z - I(\mathbf{x}))H(\phi(\mathrm{x}))d\mathbf{x}}{\int_\Omega H(\phi(\mathbf{x}))d\mathbf{x}}. \tag{101}$$

By differentiating P_i and P_o with respect to ϕ, one obtains

$$\frac{\partial P_i(z)}{\partial \phi} = \frac{\delta(\phi)}{A_i}(P_i(z) - K(z - I(\mathbf{x}))), \quad \frac{\partial P_o(z)}{\partial \phi} = \frac{\delta(\phi)}{A_o}(K(z - I(\mathbf{x})) - P_o(z)) \tag{102}$$

where $\delta(\cdot)$ is the delta function, and A_i and A_o are the areas inside and outside the segmenting curve, respectively, i.e., $A_i = \int_\Omega H(-\phi(\mathbf{x}))d\mathbf{x}$ and $A_o = \int_\Omega H(\phi(\mathbf{x}))d\mathbf{x}$. The first variation of (98) with respect to ϕ is given by

$$\begin{aligned} \frac{\partial B(\phi)}{\partial \phi} &= \frac{1}{2}\int_{\mathbb{Z}} (P_i(z)P_o(z))^{-1/2}\left(\frac{\partial P_i(z)}{\partial \phi}P_o(z) + P_i(z)\frac{\partial P_o(z)}{\partial \phi}\right) dz \\ &= \frac{1}{2}\int_{\mathbb{Z}} \left(\frac{\partial P_i(z)}{\partial \phi}\sqrt{\frac{P_o(z)}{P_i(z)}} + \frac{\partial P_o(z)}{\partial \phi}\sqrt{\frac{P_i(z)}{P_o(z)}}\right) dz. \end{aligned} \tag{103}$$

By substituting (102) into (103), we have

$$\frac{\partial B(\phi)}{\partial \phi} = \delta(\phi)S \tag{104}$$

where

$$S = \frac{B}{2}\left(\frac{1}{A_i} - \frac{1}{A_o}\right) + \frac{1}{2}\int_{\mathbb{Z}} K(z - I(\mathbf{x}))\left(\frac{1}{A_o}\sqrt{\frac{P_i(z)}{P_o(z)}} - \frac{1}{A_i}\sqrt{\frac{P_o(z)}{P_i(z)}}\right) dz. \tag{105}$$

Now we have the gradient flow of ϕ for the level set evolution minimizing (98) by introducing an artificial time parameter t:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial B(\phi)}{\partial \phi} = -\delta(\phi)S. \tag{106}$$

In (105), the first term and the second term determine the global moving direction for the entire curve and the local evolution direction, respectively.

To alleviate the degree of sensitivity to measurement noises or errors in the data, a regularizing term is added to constrain the curve length for smooth evolution. Thus, the optimal level set function is given by

$$\phi^\star = arg\ \inf_\phi \{B(\phi) + \alpha \int_\Omega \|\nabla H(\phi)\|\, d\mathbf{x}\} \tag{107}$$

where α is a user defined regularization constant ($\alpha > 0$) and ∇ denotes the gradient. The final equation of the gradient flow for level set evolution minimizing the cost functional in (107) is given by

$$\frac{\partial \phi}{\partial t} = \delta(\phi)(\alpha\kappa - S) \tag{108}$$

where the curvature of the evolving curve, κ, is given by the divergence of the gradient of the level set function $\phi : \kappa = \mathrm{div}\{\frac{\nabla\phi}{\|\nabla\phi\|}\}$. The gradient flow in (108) will converge when the curve's interior and exterior distributions, i.e., P_i and P_o, are maximally different.

For the numerical support of the level set evolution, smooth approximations of the delta function $\delta(\cdot)$ and the Heaviside step function $H(\cdot)$ can be used. Their smooth versions could be defined as given by, e.g.,

$$\delta_\epsilon(\phi) = \begin{cases} \frac{1}{2\epsilon}\left(1 + \cos\frac{\pi\phi}{\epsilon}\right), & \text{for} \quad |\phi| \leq \epsilon, \\ 0, & \text{for} \quad \text{otherwise}, \end{cases} \tag{109}$$

and

$$H_\epsilon(\phi) = \begin{cases} 1, & \text{for} \quad \phi > \epsilon, \\ 0, & \text{for} \quad \phi < \epsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\frac{\pi\phi}{\epsilon}\right), & \text{for} \quad \text{otherwise} \end{cases} \tag{110}$$

where ϵ is the user-defined parameter.

APPENDIX C

DERIVATIONS OF THE GRADIENT FLOW FOR REGION-BASED ENERGY FUNCTIONAL WITH RESPECT TO 3D POSE PARAMETERS

In this appendix, detailed computations of the gradient descent flow for the region-based energy functional E, which is used in Chapter 4, are presented. The choice of notation and terminology follows those defined in Chapter 4. The objective energy functional based on region-based active contours is given as follows:

$$E = \int_R r_o(I(\mathbf{x}), \hat{c}) d\Omega + \int_{R^c} r_b(I(\mathbf{x}), \hat{c}) d\Omega \tag{111}$$

where $r_o : \chi, \Omega \mapsto \mathbb{R}$ and $r_b : \chi, \Omega \mapsto \mathbb{R}$ are functions measuring the visual consistency of the image pixels with a statistical model over the regions R and R^c, respectively. Here, χ is the space that corresponds to photometric variable of interest. If we use the log-likelihood function, we have

$$r_o = \log(P_o), \quad r_b = \log(P_b) \tag{112}$$

where P_o and P_b are the probability density functions of the pixels inside and outside the segmenting curve, respectively, which are given as distinct Gaussian densities [81]:

$$P_o(I(\mathbf{x}), \hat{c}) = \frac{1}{\sqrt{2\pi\Sigma_{\mathrm{o}}}} \exp - \frac{(I(\mathbf{x}) - \mu_{\mathrm{o}})^2}{2\Sigma_{\mathrm{o}}}, \quad P_b(I(\mathbf{x}), \hat{c}) = \frac{1}{\sqrt{2\pi\Sigma_{\mathrm{b}}}} \exp - \frac{(I(\mathbf{x}) - \mu_{\mathrm{b}})^2}{2\Sigma_{\mathrm{b}}}. \tag{113}$$

Here Σ_o and Σ_b are variances inside and outside the curve $\hat{c}$, and given by

$$\Sigma_{\mathrm{o}} = \frac{\int_R \left(I(\mathbf{x}) - \mu_{\mathrm{o}}\right)^2 d\Omega}{\int_R d\Omega}, \quad \Sigma_{\mathrm{b}} = \frac{\int_{R^c} \left(I(\mathbf{x}) - \mu_{\mathrm{b}}\right)^2 d\Omega}{\int_{R^c} d\Omega} \tag{114}$$

where μ_{o} and μ_{b} are intensity averages:

$$\mu_{\mathrm{o}} = \frac{\int_R I(\mathbf{x}) d\Omega}{\int_R d\Omega}, \quad \mu_{\mathrm{o}} = \frac{\int_{R^c} I(\mathbf{x}) d\Omega}{\int_{R^c} d\Omega}. \tag{115}$$

For gray-scale images, $\mu_{\mathrm{o/b}}$ and $\Sigma_{\mathrm{o/b}}$ are scalars and for color images, $\mu_{\mathrm{o/b}} \in \mathbb{R}^3$ and $\Sigma_{\mathrm{o/b}} \in \mathbb{R}^{3\times3}$ are vectors and matrices. Now, we define r_o and r_b from equations above as

$$r_o = -\log(\Sigma_o) - \frac{(I(\mathbf{x}) - \mu_o)^2}{\Sigma_o}, \quad r_b = -\log(\Sigma_b) - \frac{(I(\mathbf{x}) - \mu_b)^2}{\Sigma_b}. \tag{116}$$

The partial differentials of E with respect to the 3D pose parameters, $\lambda = [\lambda_1, \ldots, \lambda_6]^T = [t_x, t_y, t_z, \omega_1, \omega_2, \omega_3]^T$, are given via the chain rule:

$$\frac{\partial E}{\partial \lambda_i} = \int_{\hat{c}} (r_o(I(\mathbf{x})) - r_b(I(\mathbf{x}))) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} + \int_R \left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega + \int_{R^c} \left\langle \frac{\partial r_b}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \tag{117}$$

where $\hat{s}$ is the arc-length parameterization of the silhouette $\hat{c}$ and $\hat{\mathbf{n}}$ is the (outward) normal to the curve at $\mathbf{x}$. Each term of (117) can be computed as follows:

- The first term in (117): Using the arc-length s of C and the direct $\frac{\pi}{2}$-rotation matrix $J = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$, we have

$$\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, J\frac{\partial \hat{c}}{\partial \hat{s}} \right\rangle d\hat{s} = \left\langle \frac{\partial \boldsymbol{\pi}(C)}{\partial \lambda_i}, J\frac{\partial \boldsymbol{\pi}(C)}{\partial s}\frac{\partial s}{\partial \hat{s}} \right\rangle d\hat{s} = \left\langle \frac{\partial \boldsymbol{\pi}(C)}{\partial \lambda_i}, J\frac{\partial \boldsymbol{\pi}(C)}{\partial s} \right\rangle ds. \tag{118}$$

  Here $\hat{\mathbf{n}} = J\hat{\mathbf{t}} = J\frac{\partial \hat{c}}{\partial \hat{s}}$ and $\hat{c} = \boldsymbol{\pi}(C)$. Since $\mathbf{X} = [X, Y, Z]^T$ is the coordinates of a point in $\mathbb{R}^3$ and $\boldsymbol{\pi}(C) = [X/Z, Y/Z]^T$, we obtain the Jacobian $\mathcal{J}$ of $\boldsymbol{\pi}(\mathbf{X})$ with respect to the spatial coordinates:

$$\mathcal{J} = \frac{\partial \boldsymbol{\pi}(C)}{\partial \mathbf{X}} = \begin{pmatrix} \frac{\partial}{\partial X}\frac{X}{Z} & \frac{\partial}{\partial Y}\frac{X}{Z} & \frac{\partial}{\partial Z}\frac{X}{Z} \\ \frac{\partial}{\partial X}\frac{Y}{Z} & \frac{\partial}{\partial Y}\frac{Y}{Z} & \frac{\partial}{\partial Z}\frac{Y}{Z} \end{pmatrix} = \frac{1}{Z^2}\begin{pmatrix} Z & 0 & -X \\ 0 & Z & -Y \end{pmatrix}. \tag{119}$$

  From (118) and (119), we get

$$\begin{aligned} \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} &= \left\langle \frac{\partial \boldsymbol{\pi}(C)}{\partial \mathbf{X}}\frac{\partial \mathbf{X}}{\partial \lambda_i}, J\frac{\partial \boldsymbol{\pi}(C)}{\partial \mathbf{X}}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds \\ &= \left\langle \mathcal{J}\frac{\partial \mathbf{X}}{\partial \lambda_i}, J\mathcal{J}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds = \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathcal{J}^T J \mathcal{J}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds \\ &= \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds \\ &= \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds = \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \|\mathbf{X}\|\mathbf{N}\sin(\theta) \right\rangle ds. \end{aligned} \tag{120}$$

  Here, the point $\mathbf{X}$ belongs to the occluding curve C. The vector $\mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}$ is the tangent to the curve C at the point $\mathbf{X}$. Since the vectors $\mathbf{t}$ and $\mathbf{X}$ belong to the tangent plane to S at $\mathbf{X}$. Thus

$\frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} = \|\mathbf{X}\|\mathbf{N}\sin(\theta)$, with $\theta = (\widehat{\mathbf{t}, \mathbf{X}})$ the angle between $\mathbf{t}$ and $\mathbf{X}$. A necessary condition for a point $\mathbf{X}$ to belong to the occluding curve is that $\langle \mathbf{X}, \mathbf{N} \rangle = 0$. For $\mathbf{X} \in C$, we have

$$\frac{\partial}{\partial s}\left\langle \mathbf{X}, \mathbf{N} \right\rangle = \underbrace{\left\langle \frac{\partial \mathbf{X}}{\partial s}, \mathbf{N} \right\rangle}_{=0 \text{ from } \mathbf{t}=\frac{\partial \mathbf{X}}{\partial s}} + \left\langle \frac{\partial \mathbf{N}}{\partial s}, \mathbf{X} \right\rangle = 0 = \left\langle d\mathbf{N}(\mathbf{t}), \mathbf{X} \right\rangle = \mathrm{II}(\mathbf{t}, \mathbf{X}). \tag{121}$$

Hence, since the second fundamental form $\mathrm{II}(\mathbf{t}, \mathbf{X}) = 0$, the vectors $\mathbf{t}$ and $\mathbf{X}$ are conjugate (see [28]). Hence, using the Euler formula, we have $K\sin^2(\theta) = \kappa_{\mathbf{X}}\kappa_{\mathbf{t}}$ where K is the Gaussian curvature, and $\kappa_{\mathbf{X}}$ and $\kappa_{\mathbf{t}}$ denote the normal curvatures in the directions $\mathbf{X}$ and $\mathbf{t}$ at $\mathbf{X} \in S$, respectively. Now, by inserting $\sin(\theta) = \sqrt{\frac{\kappa_{\mathbf{X}}\kappa_{\mathbf{t}}}{K}}$ into (120), we obtain

$$\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \frac{\|\mathbf{X}\|}{Z^3}\sqrt{\frac{\kappa_{\mathbf{X}}\kappa_{\mathbf{t}}}{K}}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds. \tag{122}$$

Thus, the first term in (117) becomes

$$\int_{\hat{c}} (r_o(I(\mathbf{x})) - r_b(I(\mathbf{x}))) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \int_{\hat{c}} (r_o(I(\mathbf{x})) - r_b(I(\mathbf{x}))) \frac{\|\mathbf{X}\|}{Z^3}\sqrt{\frac{\kappa_{\mathbf{X}}\kappa_{\mathbf{t}}}{K}}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds. \tag{123}$$

- The second term in (117):

$$\begin{aligned} \int_R \left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega &= \int_R \left\langle \frac{\partial r_o}{\partial \mu_{\mathrm{o}}}\frac{\partial \mu_{\mathrm{o}}}{\partial \hat{c}} + \frac{\partial r_o}{\partial \Sigma_o}\frac{\partial \Sigma_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \\ &= \int_R \left\langle \frac{\partial r_o}{\partial \mu_{\mathrm{o}}}\frac{\partial \mu_{\mathrm{o}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega + \int_R \left\langle \frac{\partial r_o}{\partial \Sigma_o}\frac{\partial \Sigma_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega = 0 \end{aligned} \tag{124}$$

because

$$\begin{aligned} \int_R \left\langle \frac{\partial r_o}{\partial \mu_{\mathrm{o}}}\frac{\partial \mu_{\mathrm{o}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega &= \int_R \left\langle 2\left(\frac{I(\mathbf{x}) - \mu_{\mathrm{o}}}{\Sigma_o}\right)\frac{\partial \mu_{\mathrm{o}}}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \\ &= \frac{2}{\Sigma_o}\left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \int_R (I(\mathbf{x}) - \mu_{\mathrm{o}}) d\Omega \\ &= \frac{2}{\Sigma_o}\left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \int_R I(\mathbf{x}) d\Omega - \mu_{\mathrm{o}} \int_R d\Omega \\ &= \frac{2}{\Sigma_o}\left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \underbrace{\left(\int_R d\Omega \mu_{\mathrm{o}} - \mu_{\mathrm{o}} \int_R d\Omega\right)}_{=0 \text{ from (115)}} = 0, \end{aligned} \tag{125}$$

and

$$\begin{aligned}\int_R \left\langle \frac{\partial r_o}{\partial \Sigma_o}\frac{\partial \Sigma_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega &= \int_R \left\langle \left(-\frac{1}{\Sigma_o} + \frac{1}{{\Sigma_o}^2}(I(\mathbf{x}) - \mu_{\mathrm{o}})^2\right) \frac{\partial \Sigma_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \\ &= -\frac{1}{{\Sigma_o}^2} \left\langle \frac{\partial \Sigma_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \int_R (\Sigma_o - (I(\mathbf{x}) - \mu_{\mathrm{o}})^2) d\Omega \\ &= -\frac{1}{{\Sigma_o}^2} \left\langle \frac{\partial \Sigma_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \Sigma_o \int_R d\Omega - \int_R (I(\mathbf{x}) - \mu_{\mathrm{o}})^2 d\Omega \\ &= -\frac{1}{{\Sigma_o}^2} \left\langle \frac{\partial \Sigma_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle \underbrace{\left(\Sigma_o \int_R d\Omega - \int_R d\Omega \Sigma_o\right)}_{=0 \text{ from (114)}} = 0.\end{aligned} \tag{126}$$

- The third term in (117): The third term collapses with similar approaches to the computation for the second term above by replacing r_o, Σ_o, and μ_{o} by r_b, Σ_b, and μ_{b}, respectively, i.e., we have

$$\int_{R^c} \left\langle \frac{\partial r_b}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega = 0. \tag{127}$$

Finally, by inserting r_o and r_b in (116) into (117), and using (123), (124), and (127), we have the final flow as a line integral on the curve C:

$$\frac{\partial E}{\partial \lambda_i} = \int_C \left(\log\left(\frac{\Sigma_b}{\Sigma_o}\right) + \frac{(I(\mathbf{x}) - \mu_b)^2}{\Sigma_b} - \frac{(I(\mathbf{x}) - \mu_o)^2}{\Sigma_o} \right) \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_{\mathbf{X}} \kappa_{\mathbf{t}}}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds \tag{128}$$

where the term $\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle$ can be computed for the evolution of the pose parameter λ_i which is a translation parameter ($i = 1, 2, 3$) or a rotation parameter ($i = 4, 5, 6$):

- For a translation parameter,

$$\begin{aligned}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle &= \left\langle \frac{\partial \mathbf{R}\mathbf{X}_0 + \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle \\ &= \left\langle \begin{bmatrix} \frac{\partial \lambda_1}{\partial \lambda_i} \\ \frac{\partial \lambda_2}{\partial \lambda_i} \\ \frac{\partial \lambda_3}{\partial \lambda_i} \end{bmatrix}, \mathbf{N} \right\rangle = \left\langle \begin{bmatrix} \delta_{1,i} \\ \delta_{2,i} \\ \delta_{3,i} \end{bmatrix}, \mathbf{N} \right\rangle \\ &= N_i\end{aligned} \tag{129}$$

where the Kronecker symbol $\delta_{i,j}$ was used ($\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ otherwise) and $\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$.

- For a rotation parameter,

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X_0}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \mathbf{R} \begin{bmatrix} 0 & -\delta_{6,i} & \delta_{5,j} \\ \delta_{6,i} & 0 & -\delta_{4,i} \\ -\delta_{5,i} & \delta_{4,i} & 0 \end{bmatrix} \mathbf{X}_0, \mathbf{N} \right\rangle \tag{130}$$

where $\mathbf{R} = \exp\left(\begin{bmatrix} 0 & -\lambda_6 & \lambda_5 \\ \lambda_6 & 0 & -\lambda_4 \\ -\lambda_5 & \lambda_4 & 0 \end{bmatrix} \right)$ in exponential coordinates.

REFERENCES

[1] ABIDI, M. and CHANDRA, T., "Pose estimation for camera calibration and landmark tracking," in *IEEE International Conference on Robotics and Automation*, pp. 420–426, 1990.

[2] ADALSTEINSSON, D. and SETHIAN, J., "A fast level set method for propagating interfaces," *Journal of Computational Physics*, vol. 118, no. 2, pp. 269–277, 1995.

[3] ALATTAR, A. and JANG, J., "A new stereo correspondence method for snake-based object segmentation," in *IEEE International Conference on Image Processing*, vol. 3, pp. 381–384, 2007.

[4] ARULAMPALAM, M., MASKELL, S., GORDON, N., and CLAPP, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[5] BARTESAGHI, A. and SAPIRO, G., "Tracking of moving objects under severe and total occlusions," in *IEEE International Conference on Image Processing*, vol. 1, pp. 301–304, 2005.

[6] BAYRO-CORROCHANO, E. and ORTEGÓN-AGUILAR, J., "Lie algebra approach for tracking and 3D motion estimation using monocular vision," *Image and Vision Computing*, vol. 25, no. 6, pp. 907–921, 2007.

[7] BIRCHFIELD, S. and RANGARAJAN, S., "Spatiograms versus histograms for region-based tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1158–1163, 2005.

[8] BISHOP, G., WELCH, G., and ALLEN, B., "Tracking: Beyond 15 minutes of thought." in SIGGRAPH Course 11, 2001.

[9] BLAKE, A. and ISARD, M., *Active Contours*. Springer, 1998.

[10] BREGLER, C. and MALIK, J., "Tracking people with twists and exponential maps," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8–15, 2000.

[11] CAI, Q. and AGGARWAL, J., "Tracking human motion in structured environments using a distributed-camera system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1241–1247, 2002.

[12] CASELLES, V., KIMMEL, R., and SAPIRO, G., "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[13] CHAN, T. and VESE, L., "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[14] CHEN, K., LAI, C., HUNG, Y., and CHEN, C., "An adaptive learning method for target tracking across multiple cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[15] CHEN, Y., HUANG, T., and RUI, Y., "Parametric contour tracking using unscented Kalman filter," in *IEEE International Conference on Image Processing*, vol. 3, pp. 613–616, 2002.

[16] CHILGUNDE, A., KUMAR, P., RANGANATH, S., and HUANG, W., "Multi-camera target tracking in blind regions of cameras with non-overlapping fields of view," in *British Machine Vision Conference*, pp. 397–406, 2004.

[17] COMANICIU, D., RAMESH, V., and MEER, P., "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 564–575, 2003.

[18] CONAIRE, C., O'CONNOR, N., and SMEATON, A., "An improved spatiogram similarity measure for robust object localisation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 1069–1072, 2007.

[19] CREMERS, D., KOHLBERGER, T., and SCHNÖRR, C., "Nonlinear shape statistics in Mumford-Shah based segmentation," in *European Conference on Computer Vision*, pp. 516–518, 2002.

[20] CREMERS, D., ROUSSON, M., and DERICHE, R., "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 195–215, 2007.

[21] DA FONTOURA COSTA, L. and CESAR, R., *Shape Analysis and Classification: Theory and Practice*. CRC Press, 2001.

[22] DALAL, N. and TRIGGS, B., "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2005.

[23] DAMBREVILLE, S., NIETHAMMER, M., YEZZI, A., and TANNENBAUM, A., "A variational framework combining levelsets and thresholding," in *British Machine Vision Conference*, 2007.

[24] DAMBREVILLE, S., RATHI, Y., and TANNENBAUM, A., "Tracking deformable objects with unscented Kalman filtering and geometric active contours," in *American Control Conference*, vol. 6, pp. 2856–2861, 2006.

[25] DAMBREVILLE, S., SANDHU, R., YEZZI, A., and TANNENBAUM, A., "A geometric approach to joint 2D region-based segmentation and 3D pose estimation using a 3D shape prior," *SIAM Journal on Imaging Sciences*, vol. 3, no. 1, pp. 110–132, 2010.

[26] DEUTSCHER, J., BLAKE, A., and REID, I., "Articulated body motion capture by annealed particle filtering," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 126–133, 2000.

[27] DHOME, M., RICHETIN, M., LAPRESTE, J., and RIVES, G., "Determination of the attitude of 3D objects from a single perspective view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1265–1278, 1989.

[28] DO CARMO, M., *Differential Geometry of Curves and Surfaces*, vol. 2. Prentice-Hall, 1976.

[29] DOUCET, A., GODSILL, S., and ANDRIEU, C., "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[30] DRUMMOND, T. and CIPOLLA, R., "Real-time visual tracking of complex structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 932–946, 2002.

[31] DUDA, R., HART, P., and STORK, D., *Pattern Classification*. Wiley-Interscience, 2000.

[32] ELGAMMAL, A., DURAISWAMI, R., and DAVIS, L., "Probabilistic tracking in joint feature-spatial spaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 781–788, 2003.

[33] FREEDMAN, D. and ZHANG, T., "Active contours for tracking distributions," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 518–526, 2004.

[34] GELAUTZ, M. and MARKOVIC, D., "Recognition of object contours from stereo images: An edge combination approach," in *International Symposium on 3D Data Processing, Visualization and Transmission*, pp. 774–780, 2004.

[35] GEVERS, T. and SMEULDERS, W., "Color based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[36] GILKS, W. and BERZUINI, C., "Following a moving target-Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[37] GORDON, N., SALMOND, D., and SMITH, A., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F on Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[38] HA, J., JOHNSON, E., and TANNENBAUM, A., "Real-time visual tracking using geometric active contours for the navigation and control of UAVs," in *American Control Conference*, pp. 365–370, 2007.

[39] HAN, B., COMANICIU, D., ZHU, Y., and DAVIS, L., "Incremental density approximation and kernel-based Bayesian filtering for object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 638–644, 2004.

[40] HARTLEY, R. and ZISSERMAN, A., *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[41] HORN, B., *Robot vision*. The MIT Press, 1986.

[42] HU, W., TAN, T., WANG, L., and MAYBANK, S., "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, 2004.

[43] HUANG, T. and RUSSELL, S., "Object identification in a Bayesian context," in *International Joint Conference on Artificial intelligence*, pp. 1276–1282, 1997.

[44] ISARD, M. and BLAKE, A., "Condensation-conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[45] JACKSON, J., YEZZI, A., and SOATTTO, S., "Tracking deformable moving objects under severe occlusions," in *IEEE Conference on Decision and Control*, pp. 2990–2995, 2004.

[46] JAGODZINSKI, M., KLEEMANN, V., ANGELE, P., SCHÖNHAAR, V., ISELBORN, K., MALL, G., and NERLICH, M., "Experimental and clinical assessment of the accuracy of knee extension measurement techniques," *Knee Surgery, Sports Traumatology, Arthroscopy*, vol. 8, no. 6, pp. 329–336, 2000.

[47] JOHNSON, E., PROCTOR, A., HA, J., and TANNENBAUM, A., "Visual search automation for unmanned aerial vehicles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 219–232, 2005.

[48] JULIER, S. and UHLMANN, J., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–420, 2004.

[49] KAILATH, T., "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, 1967.

[50] KALMAN, R., "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[51] KANADE, T. and OKUTOMI, M., "A stereo matching algorithm with an adaptive window: Theory and experiment," in *IEEE International Conference on Robotics and Automation*, pp. 1088–1095, 1991.

[52] KASS, M., WITKIN, A., and TERZOPOULOS, D., "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[53] KIM, J., FISHER III, J., YEZZI, A., CETIN, M., and WILLSKY, A., "A nonparametric statistical method for image segmentation using information theory and curve evolution," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1486–1502, 2005.

[54] KOHLI, P., RIHAN, J., BRAY, M., and TORR, P., "Simultaneous segmentation and pose estimation of humans using dynamic graph cuts," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 285–298, 2008.

[55] KWON, J. and PARK, F., "Visual tracking via particle filtering on the affine group," *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 198–217, 2010.

[56] LATECKI, L. and MIEZIANKO, R., "Object tracking with dynamic template update and occlusion detection," in *IEEE International Conference on Pattern Recognition*, vol. 1, pp. 556–560, 2006.

[57] LEE, J., KARASEV, P., and TANNENBAUM, A., "Range based object tracking and segmentation," in *IEEE International Conference on Image Processing*, pp. 4641–4644, 2010.

[58] LEE, J., KARASEV, P., ZHU, L., and TANNENBAUM, A., "Human body tracking and joint angle estimation from mobile-phone video for clinical analysis," in *IAPR Conference on Machine Vision Applications*, pp. 475–478, 2011.

[59] LEE, J., LANKTON, S., and TANNENBAUM, A., "Object tracking and target reacquisition based on 3D range data for moving vehicles," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2912–2924, 2011.

[60] LEE, J., SANDHU, R., and TANNENBAUM, A., "Monte Carlo sampling for visual pose tracking," in *IEEE International Conference on Image Processing*, pp. 509–512, 2011.

[61] LEE, J., SANDHU, R., and TANNENBAUM, A., "Particle filters and occlusion handling for rigid 2D-3D pose tracking," *IEEE Transactions on Image Processing*, 2011. submitted for publication.

[62] LEE, M. and COHEN, I., "A model-based approach for estimating human 3D poses in static images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 905–916, 2006.

[63] LEVENTON, M., GRIMSON, W., and FAUGERAS, O., "Statistical shape influence in geodesic active contours," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 316–323, 2000.

[64] LI, P., ZHANG, T., and B., M., "Unscented Kalman filter for visual curve tracking," *Image and Vision Computing*, vol. 22, no. 2, pp. 157–164, 2004.

[65] LING, H. and OKADA, K., "Diffusion distance for histogram comparison," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 246–253, 2006.

[66] LOUTAS, E., PITAS, I., and NIKOU, C., "Probabilistic multiple face detection and tracking using entropy measures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 128–135, 2004.

[67] LOWE, D., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[68] MA, Y., SOATTO, S., KOSECKA, J., and SASTRY, S., *An Invitation to 3D Vision*. Spinger, 2003.

[69] MALCOLM, J., RATHI, Y., YEZZI, A., and TANNENBAUM, A., "Fast approximate surface evolution in arbitrary dimension," in *SPIE Medical Imaging*, vol. 6914, pp. 69144C–69144C–9, 2008.

[70] MARCHAND, É., BOUTHEMY, P., and CHAUMETTE, F., "A 2D-3D model-based approach to real-time visual tracking," *Image and Vision Computing*, vol. 19, no. 13, pp. 941–955, 2001.

[71] MARKOVIC, D. and GELAUTZ, M., "Experimental combination of intensity and stereo edges for improved snake segmentation," *Pattern Recognition and Image Analysis*, vol. 17, no. 1, pp. 131–135, 2007.

[72] MARR, D. and POGGIO, T., "Cooperative computation of stereo disparity," *Science*, vol. 194, no. 4262, pp. 283–287, 1976.

[73] MICHAILOVICH, O., RATHI, Y., and TANNENBAUM, A., "Image segmentation using active contours driven by the Bhattacharyya gradient flow," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2787–2801, 2007.

[74] MOESLUND, T. and GRANUM, E., "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[75] MUSSO, C., OUDJANE, N., and LEGLAND, F., "Improving regularised particle filters," in *Sequential Monte Carlo Methods in Practice* (DOUCET, A., DE FREITAS, N., and GORDON, N., eds.), pp. 247–271, Springer, 2001.

[76] NGUYEN, H. and SMEULDERS, A., "Fast occluded object tracking by a robust appearance filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1099–1104, 2004.

[77] NIETHAMMER, M., TANNENBAUM, A., and ANGENENT, S., "Dynamic active contours for visual tracking," *IEEE Transactions on Automatic Control*, vol. 51, no. 4, pp. 562–579, 2006.

[78] OSHER, S. and SETHIAN, J., "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[79] PAN, Y., BIRDWELL, J., and DJOUADI, S., "Efficient implementation of the Chan-Vese models without solving PDEs," in *IEEE Workshop on Multimedia Signal Processing*, pp. 350–354, 2006.

[80] PARAGIOS, N. and DERICHE, R., "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, 2000.

[81] PARAGIOS, N. and DERICHE, R., "Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1/2, pp. 249–268, 2002.

[82] PEREZ, P., HUE, C., VERMAAK, J., and GANGNET, M., "Color-based probabilistic tracking," in *European Conference on Computer Vision*, vol. I, pp. 661–675, 2002.

[83] PETERFREUND, N., "Robust tracking of position and velocity with Kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, 1999.

[84] PETERFREUND, N., "The velocity snake: Deformable contour for tracking in spatio-velocity space," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 346–356, 1999.

[85] PHUNG, S., BOUZERDOUM, A., and CHAI, D., "Skin segmentation using color pixel classification: Analysis and comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 148–154, 2005.

[86] RATHI, Y., VASWANI, N., and TANNENBAUM, A., "A generic framework for tracking using particle filter with dynamic shape prior," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1370–1382, 2007.

[87] RATHI, Y., VASWANI, N., TANNENBAUM, A., and YEZZI, A., "Particle filtering for geometric active contours with application to tracking moving and deforming objects," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2–9, 2005.

[88] RATHI, Y., VASWANI, N., TANNENBAUM, A., and YEZZI, A., "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, 2007.

[89] RIKLIN-RAVIV, T., KIRYATI, N., and SOCHEN, N., "Prior-based segmentation by projective registration and level sets," in *IEEE International Conference on Computer Vision*, vol. 1, pp. 204–211, 2005.

[90] RISTIC, B., ARULAMPALAM, S., and GORDON, N., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[91] ROOZ, N., JOHNSON, E., WU, A., CHRISTMANN, C., HA, J., CHOWDHARY, G., SOBERS, M., KANNAN, S., PICKELL, W., CHRISTOPHERSEN, H., TANNENBAUM, A., LEE, J., HUR, J., KIMBRELL, S., GATES, H., ANDRUS, B., and PROCTOR, A., "Experience with highly automated unmanned aircraft performing complex missions," in *AIAA Guidance, Navigation, and Control Conference*, 2009.

[92] ROSENHAHN, B., BROX, T., and WEICKERT, J., "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 243–262, 2007.

[93] ROSENHAHN, B., PERWASS, C., and SOMMER, G., "Pose estimation of 3D free-form contours," *International Journal of Computer Vision*, vol. 62, no. 3, pp. 267–289, 2005.

[94] SABETI, L., PARVIZI, E., and WU, Q., "Visual tracking using color cameras and time-of-flight range imaging sensors," *Journal of Multimedia*, vol. 3, no. 2, pp. 28–36, 2008.

[95] SANDHU, R., DAMBREVILLE, S., and TANNENBAUM, A., "Point set registration via particle filtering and stochastic dynamics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1459–1473, 2009.

[96] SANDHU, R., DAMBREVILLE, S., YEZZI, A., and TANNENBAUM, A., "Non-rigid 2D-3D pose estimation and 2D image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 786–793, 2009.

[97] SANDHU, R., GEORGIOU, T., and TANNENBAUM, A., "Tracking with a new distribution metric in a particle filtering framework," in *IST/SPIE Symposium on Electronic Imaging*, vol. 6813, 2008.

[98] SCHARSTEIN, D. and SZELISKI, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2002.

[99] SCHARSTEIN, D. and SZELISKI, R., "High-accuracy stereo depth maps using structured light," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 195–202, 2003.

[100] SCHMALTZ, C., ROSENHAHN, B., BROX, T., CREMERS, D., WEICKERT, J., WIETZKE, L., and SOMMER, G., "Region-based pose tracking," *Pattern Recognition and Image Analysis*, pp. 56–63, 2007.

[101] SEBASTIAN, P., VOON, Y., and COMLEY, R., "The effect of colour space on tracking robustness," in *IEEE Conference on Industrial Electronics and Applications*, pp. 2512–2516, 2008.

[102] SETHIAN, J., *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

[103] SHAN, C., TAN, T., and WEI, Y., "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, no. 7, pp. 1958–1970, 2007.

[104] SHI, J. and TOMASI, C., "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[105] SHI, Y. and KARL, W., "Real-time tracking using level sets," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 34–41, 2005.

[106] SIMON, D., *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[107] SONG, B. and CHAN, T., "A fast algorithm for level set based optimization," Tech. Rep. CAM02-68, UCLA, CAM, 2002.

[108] SRIVASTAVA, A., "Bayesian filtering for tracking pose and location of rigid targets," in *SPIE Signal Processing, Sensor Fusion, and Target Recognition*, vol. 4052, pp. 160–171, 2000.

[109] SULLIVAN, J. and RITTSCHER, J., "Guiding random particles by deterministic search," in *IEEE International Conference on Computer Vision*, vol. 1, pp. 323–330, 2001.

[110] TALUKDER, A. and MATTHIES, L., "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, pp. 3718–3725, 2004.

[111] TANNENBAUM, A., "Three snippets of curve evolution theory in computer vision," *Mathematical and Computer Modelling*, vol. 24, pp. 103–120, 1996.

[112] TERZOPOULOS, D. and SZELISKI, R., "Tracking with Kalman snakes," in *Active Vision* (BLAKE, A. and YUILLE, A., eds.), pp. 3–20, The MIT Press, 1993.

[113] VERMAAK, J., LAWRENCE, N., and PEREZ, P., "Variational inference for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 773–780, 2003.

[114] WAN, E. and VAN DER MERWE, R., "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.

[115] WEI, J., WANG, S., CHEN, L., and GUAN, T., "Adaptive stereo video object segmentation based on depth and spatio-temporal information," in *World Congress on Computer Science and Information Engineering*, pp. 140–144, 2009.

[116] WHITAKER, R., "A level-set approach to 3D reconstruction from range data," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 203–231, 1998.

[117] WIMMER, M., RADIG, B., and BEETZ, M., "A person and context specific approach for skin color classification," in *IEEE International Conference on Pattern Recognition*, vol. 2, pp. 39–42, 2006.

[118] YANG, C., DURAISWAMI, R., and DAVIS, L., "Fast multiple object tracking via a hierarchical particle filter," in *IEEE International Conference on Computer Vision*, vol. 1, pp. 212–219, 2005.

[119] YEZZI, A. and SOATTO, S., "Stereoscopic segmentation," *International Journal of Computer Vision*, vol. 53, no. 1, pp. 31–43, 2003.

[120] YEZZI, A., TSAI, A., and WILLSKY, A., "A fully global approach to image segmentation via coupled curve evolution equations," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1, pp. 195–216, 2002.

[121] YEZZI, A. and SOATTO, S., "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 153–167, 2003.

[122] YEZZI, A. and SOATTO, S., "Structure from motion for scenes without features," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 525–532, 2003.

[123] YILMAZ, A., JAVED, O., and SHAH, M., "Object tracking: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, pp. 1–45, 2006.

[124] YILMAZ, A., LI, X., and SHAH, M., "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531–1536, 2004.

[125] ZHANG, T. and FREEDMAN, D., "Tracking objects using density matching and shape priors," in *IEEE International Conference on Computer Vision*, pp. 1056–1062, 2003.

[126] ZHU, S. and YUILLE, A., "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.